НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ ЗАХИСТУ ІНФОРМАЦІЇ

«До захисту допущено»
В.о. завідувача кафедрою
_____ М.М.Савчук
(підпис) (ініціали, прізвище)
“ 11 ”червня 2019 р.

Дипломна робота

на здобуття ступеня бакалавра

з напряму підготовки 6.040301 «Прикладна математика»
(код і назва)

на тему: Обчислення оцінок стійкості шифрів, побудованих на методах перемішування карт

Виконав (-ла): студент (-ка) 4 курсу, групи ФІ-52
(шифр групи)

Тузовська Марія Андріївна

Керівник Фесенко А.В. к.ф.-м.н., ст.. викладач (підпис)
(посада, науковий ступінь, вчене звання, прізвище та ініціали)

Консультант _____ _____ _____ (підпис)
(назва розділу) (посада, вчене звання, науковий ступінь, прізвище, ініціали)

Рецензент к.ф.-м.н., доцент Южакова Г.О. (підпис)
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали)

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.
Студент _____
(підпис)

Київ – 2019року

Національний технічний університет України
«Київський політехнічний інститут
імені Ігоря Сікорського»
Фізико-технічний інститут

Кафедра математичних методів захисту інформації

Рівень вищої освіти – перший (бакалаврський)

Напрям підготовки - 6.040301 «Прикладна математика»

ЗАТВЕРДЖУЮ
В.о. завідувача кафедрою

М.М.Савчук

_____
(підпис) (ініціали, прізвище)

«13» лютого 2019р.

ЗАВДАННЯ
на дипломну роботу студенту

Тузовської Марії Андріївни
(прізвище, ім'я, по батькові)

1. Тема роботи: Обчислення оцінок стійкості шифрів, побудованих на методах перемішування карт,

керівник роботи Фесенко А.В., к.ф.-м.н., ст.. викладач,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від 27.05.2019 р. № 1414-с

2. Термін подання студентом роботи 16.09.2019

3. Вихідні дані до роботи _____
_____
_____
_____

4. Зміст роботи: Зроблено огляд класичних та сучасних підходів криптографічного перетворення інформації за допомогою шифрів, які використовують квантові перемішування.

5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо) _____
_____

_____

_____

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання _____

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка

Студент _____ (підпис) _____ (ініціали, прізвище)

Керівник роботи _____ (підпис) _____ (ініціали, прізвище)

РЕФЕРАТ

Кваліфікаційна робота містить: 79 стор., 4 рисунки, 39 джерел. Метою дипломної роботи є обчислення оцінок стійкості шифрів, побудованих на методах перемішування карт.

Об'єктом дослідження є процес криптографіного перетворення інформації за допомогою шифрів, які використовують методи перемішування карт.

Предметом дослідження є стійкість шифрів, які використовують методи перемішування карт.

В роботі зроблено огляд класичних та сучасних підходів криптографіного перетворення інформації за допомогою шифрів, які використовують карткові перемішування. Узагальнено схему шифрування Mix-And-Cut та запропоновано новий шифр на основі Mix-And-Cut. Досліджені властивості такого шифру та обчислені оцінки стійкості. Також, у роботі запропонована нова атака на шифр Swap-Or-Not.

AES, SWAP-OR-NOT, ПЕРЕМІШУВАННЯ ТОРПА, MIX-AND-CUT, MIX-AND-COIN, БЛОЧНІ ШИФРИ, ШИФРУВАННЯ ЗІ ЗБЕРЕЖЕННЯМ ФОРМАТУ

РЕФЕРАТ

Квалификационная робота содержит: 79 страницы, 4 рисунка, 39 источников. Целью дипломной работы является подсчет оценок стойкости шифров, что даст возможность качественно использовать структуру таких схем для криптографии.

Объектом исследования является процесс криптографического преобразования информации с помощью шифров, которые используют карточные схемы подстановок.

Предметом исследования является стойкость шифров, которые используют методы подстановок карт.

В работе проведен обзор классических и современных подходов криптографического преобразования информации с помощью шифров, которые используют схемы подстановок карт. Обобщена схема шифрования Mix-And-Cut и предложен новый шифр на основе Mix-And-Cut. Исследованы свойства такого шифра и посчитаны оценки стойкости. В работе предложена новая атака на шифр Swap-Or-Not.

AES, SWAP-OR-NOT, ПЕРЕМЕШИВАНИЕ ТОРПА, MIX-AND-CUT, MIX-AND-COIN, БЛОКОВЫЕ ШИФРЫ, ШИФРОВАНИЕ СОХРАНЯЮЩЕЕ ФОРМАТ

ABSTRACT

The thesis contains: 79 pages, 4 figures, 39 sources.

In this thesis, a detailed analysis was made of the calculation evaluation of ciphers based on the methods of cards shuffle.

The object is the process of cryptographic conversion of information, using ciphers that based on card shuffle.

The subject is the security of ciphers that use card shuffling methods.

In this thesis, a detailed analysis was made of classical and modern approaches of cryptographic transformation of information, used ciphers that based on card shuffle. Generalized Mix-And-Cut cipher and introduced new cipher based on Mix-And-Cut. Reviewed properties of this cipher and calculated its security. Also, in this work proposed new attack on Swap-Or-Not cipher.

AES, SWAP-OR-NOT, THORP SHUFFLE, MIX-AND-CUT, MIX-AND-COIN, BLOCK CIPHERS, FORMAT PRESERVING ENCRYPTION

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

FPE (анг. format-preserving encryption) — шифрування зі збереженням формату

AES (анг. Advanced Encryption Standart) — симетричний алгоритм блочного шифрування

PRF (анг. pseudorandom function family) — сімейство псевдовипадкових функцій

PRS (анг. pseudorandom separator) — псевдовипадковий розділювач

PRP (анг. pseudorandom permutation) — псевдовипадкова перестановка

CCA (анг. chosen ciphertext attack) — атака на основі обраного шифрованого тексту

CPA (анг. chosen plaintext attack) — атака на основі обраного відкритого тексту

C (анг. Coin) — функція підкидання монетки, рівноймовірна

$\oplus$ — операція побітового додавання (за модулем 2)

mod — операція отримення остачі від ділення

div — операція отримання цілої частини при діленні

ВСТУП

Актуальність дослідження полягає в задачі шифрування даних зі збереженням формату невеликої довжини вхідних даним. На сьогодні дуже багато сучасних систем потребують методів шифрування, які б забезпечили конфіденційність, цілісність даних, при цьому, шифрування було б швидким і забезпечило збереження формату відкритого тексту у зашифрованому.

Необхідність використовувати схеми та алгоритми, які зберігають довжину повідомлення або формат на сьогодні є актуальним питанням. Вони наявні у багатьох додатках та процесах: в процесах охорони здоров'я, в системахобробки авіаліній, у мобільних додатках, у великих базах даних, у веб-транзакціях, а також у наданні організаціям перенесення конфіденційних даних у хмару, зберігаючи повний контроль над своїми даними.

Об'єктом дослідження є процес криптографіного перетворення інформації за допомогою шифрів, які використовують методи перемішування карт.

Предметом дослідження є стійкість шифрів, які використовують методи перемішування карт.

Метою дослідження є обчислення оцінок стійкості шифрів, побудованих на методах перемішування карт. Для досягнення мети необхідно розв'язати наступну **задачу дослідження:** оцінити стійкість деяких шифрів, побудованих на методах перемішування карт, узагальнити конструкцію шифрів Swap-Or-Not та Mix-And-Cut та оцінити рівень її стійкості. Для розв'язання поставленої задачі необхідно вирішити настінні **завдання:**

1) провести огляд опублікованих джерел за тематикою дослідження;

2) побудувати класифікацію існуючих методів перемішування карт та дослідити їх властивості;

3) описати моделі шифрування на схемах перемішування карт, визначити їх перевагу чи недоліки;

4) проаналізувати шифри Торпа, Swap-Or-Not, Mix-And-Cut та переглянути їх оцінки стійкості;

5) побудувати новий шифр Mix-And-Coin;

6) обчислити оцінки стійкості для нерозрізненості побудованого шифру.

При розв'язанні поставлених завдань використовувались такі *методи дослідження*: методи лінійної та абстрактної алгебри, теорії ймовірностей, математичної статистики, комбінаторного аналізу, теорії складності алгоритмів.

Наукова новизна результатів полягає в побудові шифру Mix-And-Coin який є узагальненням шифру Mix-And-Cut, обчисленні оцінок стійкості побудованого шифру; уточненні оцінок стійкості шифру та Swap-Or-Not завдяки новій, побудованій в цій роботі, атаці.

Апробація результатів та публікації. Результати цієї роботи були частково представлені на XVII Науково-практичній конференції студентів, аспірантів та молодих вчених "Теоретичні і прикладні проблеми фізики, математики таінформатики"(26-27 квітня 2019р., м. Київ).

1 ТЕОРЕТИЧНІ ВІДОМОСТІ

У цьому розділі визначені основні теоретичні результати і визначення шифрування зі збереженням формату та шифрування повідомлень невеликої довжини. Узагальнені карткові перемішування. Наведені алгоритми карткових перемішувань.

1.1 Криптографічні методи захисту інформації

Проблеми конфіденційності, цілісності та автентичності інформації при передачі по відкритим каналам вирішуються за допомогою криптографічних методів захисту інфомації. Криптографія вивчає способи перетворення інформації з метою її захисту від несанкціонованого доступу шляхом шифрування даних. Потреба в забезпеченні надійного шифрування призвела до створення великої кількості алгоритмів шифрування та розшифровування даних різними способами, в залежності від поставленої задачі.

Шифрування являє собою спосіб зміни повідомлення або іншого документа, що забезпечує спотворення (приховання) його вмісту. Шифрувати можна не тільки текст, але і різні комп'ютерні файли – від файлів баз даних і текстових процесорів до файлів зображень. Розшифровування — зворотній процес шифрування.

Алгоритми шифрування і розшифровування повідомлень широко застосовуються в сучасному світі для приховання конфіденційної і комерційної інформації від несанкціонованого використання сторонніми особами. Головним принципом у них є умова, що особа яка приймає повідомлення, заздалегідь знає алгоритм шифрування, а також ключ до

повідомлення, без якого інформація є всього лише набір символів, що не мають сенсу.

Для подальшої роботи розглянемо алгебраїчну модель симетричного шифру, запропоновану Шеноном у 1949 році [1], що далі буде використовуватися для опису блочних шифрів та схем шифрування зі збереженням формату.

Означення 1.1. Алгебраїчна модель симетричного (детермінованого, скінченного) шифру є універсальною алгеброю

$$\mathcal{A} =< \mathbb{K}, \mathbb{X}, \mathbb{Y}, E >$$

з системою носіїв ($\mathbb{K}$, $\mathbb{X}$, $\mathbb{Y}$), де $\mathbb{K}$, $\mathbb{X}$, $\mathbb{Y}$ — непорожні скінченні множини, і однією алгебраїчною операцією, яка задовольняє умовам:

$$E\colon \mathbb{K} \times \mathbb{X} \to \mathbb{Y}$$

1. $E\colon \mathbb{K} \times \mathbb{X} \to \mathbb{Y}$ — сюр'єкція;

2. для довільних значень $k \in \mathbb{K}$ та $x_1, x_2 \in \mathbb{X}$ виконується співвідношення: $x_1 \neq x_2 \Rightarrow E(k, x_1) \neq E(k, x_2)$.

Скінченні множини $\mathbb{K}$, $\mathbb{X}$ та $\mathbb{Y}$ називають відповідно (простором) множиною (особистих) ключів, (простором) множиною відкритих текстів (або повідомлень) та (простором) множиною шифрованих текстів (або шифротекстів), а відображення E — функцією шифрувння шифру $\mathcal{A}$.

Отже, введена математична модель, яка відображає основні властивості реальних симетричних шифрів. В силу цього, будемо визначати симетричний шифр на алгебраїчній моделі (див. означення 1.1), яку далі в роботі будемо називати просто шифром, як і довільну універсальну алгебру, що задовольняє умовам означення — також будемо називати шифром.

Означення 1.2. Блочним шифром будемо називати алгебраїчну модель шифрування (див. означення 1.1) з характеристиками:

1. Шифрування повідомлення виконується на множинах $\{0, 1\}^k \times \{0, 1\}^n \to \{0, 1\}^n$;

2. Для кожного $k \in \mathbb{K}$, E_k — бієктивна функція (перестановка) на множині n-бітових блоків;

3. Для кожного $k \in \mathbb{K}$, для кожного $x \in \mathbb{X}$ та для кожного $y \in \mathbb{Y}$: $E_k(x) = y$ і $D_k(y) = x$;

На сьогоднішній день розроблено досить багато стійких блокових шифрів. Практично всі алгоритми використовують для перетворень певний набір бієктивних (оборотних) математичних перетворень.

Характерною особливістю блочних шифрів є той факт, що в ході своєї роботи вони перетворюють блок вхідної інформації фіксованої довжини і отримують результуючий блок того ж обсягу, але недоступний для прочитання стороннім особам, що не володіють ключем.

Блочні шифри є основою, на якій реалізовані практично всі криптосистеми. Властивість блокових шифрів — це перевага у швидкості роботи, на відміну від асиметричних криптоалгоритмів, які є повільними за своєю природою. Відсутність статистичної кореляції між бітами вихідного потоку блочного шифру використовується для обчислення контрольних сум пакетів даних і в хешуванні паролів.

Розглянемо відомий блочний шифр AES (Advanced Encryption Standard). Відкритий в 1997 році Національним Інститутом Інформаційних Стандартів і Технологій США (NIST). Програми з розробки AES були виконані в три проміжних етапи міжнародного конкурсу. Тоді новий стандарт мав задоволити потреби:

— стійкість не менше ніж у 3DES;

— швидкість шифрування більше ніж в 3DES;

— прозору структуру;

— ефективну реалізацію на платформі Pentium Pro;

— ефективну апаратну реалізацію.

Загальні характеристики блочного шифру AES.

1. AES зашифровує та розшифровує 128-бітові блоки даних.

2. AES дозволяє використовувати три різні ключі довжиною 128, 192 або 256 біт (в залежності від довжини ключа шифру називають AES-128, AES-192 або AES-256).

3. Від розміру ключа залежить число раундів шифрування: довжина 128 біт — 10 раундів, 192 біти — 12 раундів, 256 біт — 14 раундів.

Криптографія виникла як спосіб приховання інформації і це не означає, що всі шифри є бездоганними, а інформація цілком захищена. Із ходом розвинення науки, виникли способи обходу шифрів, тобто реалізація атак на криптографічні примітиви. Введемо означення, для визначення стійкості шифрів до різного типу атак.

Означення 1.3. Атака на основі обраного відкритого тексту (анг. Chosen-plaintext attack) — один з основних способів криптоаналітичного отримання конфіденційних даних. Криптоаналітик володіє певною кількістю відкритих текстів та відповідних їм шифротекстів. Має можливість зашифрувати обрані ним відкриті тексти.

Означення 1.4. Атака на основі підібраного шифротексту (англ. Chosen-ciphertext attack) — криптографічна атака, при якій криптоаналітик збирає інформацію про систему шифрування через підбір зашифрованого тексту і отримання відповідного відкритого тексту. Його завдання відновити ключ. Як правило, криптоаналітик може скористуватися пристроєм розшифрування один або декілька разів.

Оцінки стійкості криптографічних систем захисту інформації є досить актуальним у наш час, коли існує багато схем шифрування і постає задача вибору дійсно ефективного та стійкого шифру для підвищення загального рівня конфіденційності, тож для вибору стійкої криптографічної системи, обчислюються оцінки стійкості до атак зловмисника (див. означення 1.3, 1.4, та інші).

1.2 Шифрування зі збереженням формату

За деяких обставин, при роботі з застарілим середовищем, необхідно зашифровувати текст в тому ж форматі і довжині, використовуючи той же набір символів, що і оригінальний відкритий текст. Наприклад, номер платіжного рахунку (PAN) складається з довжини до 19 десяткових цифр, причому перші 6 цифр використовуються для ідентифікації схеми оплати та банку, який видав рахунок, а остання цифра використовується як контрольна сума (використовується для перевірки цілісності даних при передачі чи збереженні). Обчислювальні процеси оплати картою можуть шифрувати ці номери таким чином, що зашифрований текст зберігає деякі, або всі вихідні атрибути відкритого тексту так, що системи, призначені для обробки відкритого тексту, також можуть обробляти зашифрований текст. Такий вид шифрування у криптографії називається шифрування, що зберігає формат.

Означення 1.5. Шифрування зі збереженням формату (анг. format preserving encryption) — це спосіб шифрування (зазвичай, це блочні симетричні криптосистеми), при якому шифровані (вихідні) дані (текст) мають ідентичний формат (атрибути), що і вхідні дані.

Задача шифрування дійсного номера кредитної картки в дійсний номер кредитної картки була відома протягом деякого часу, але їй бракувало повного розв'язку.

Шифрування зі збереженням формату [2] дозволяє шифрувати дані з нестандартними форматами, які не підходять для модифікації (наприклад: інформація, яка передається у мережі, відмінна від IP, або зберігається у застарілих базах данних). Розроблені давно комунікаційні протоколи, які використовуються в задачах передачі даних — несумісні з сучасною безпекою, основаною на IP і дуже дорогі для заміни. Шифрування зі збереженням формату може потенціально забезпечити

безпеку у застарілих системах критично важливої інфраструктури, які були розроблені з переліком вимог безпеки, несумісних зі стандартною технологією шифрування.

Протягом останніх кількох років — шифрування, що зберігає формат, виникло як корисний інструмент у прикладній криптографії. Мета полягає в наступному [3]: використовуючи модель блочного шифрування (див. означення 1.2), детерміновано шифрувати відкритий текст X у зашифрований Y, який має той же формат, що і X. Метод шифрування приймає рядки символів як вхідний текст і створює вихід (шифрований текст), такої ж довжини, що і вхідні дані, використовуючи той же набір символів. Для наведеного вище прикладу це дозволить процесу шифрування банківської картки шифрувати PAN і зберігати той самий формат і структуру в зашифрованому тексті.

Блочное шифрування n-бітового рядка (наприклад, AES) технічно і є шифруванням зі збереженням формату на множині $\{0, 1, ..., 2^{n-1}\}$. Якщо необхідно зберегти формат повідомлення на одній зі стандартних множин (тобто, де довжина повідомлення $n = 128, 192, 256$ бітів), можна використовувати блочне шифрування необхідного розміру.

Режим роботи блочного шифру (або просто режим) — це алгоритм криптографічного перетворення даних, заснованих на блочному шифрі. Затверджені раніше режими для шифрування є перетвореннями послідовностей входу на двійкові дані, тобто входи і виходи режимів є бітовими рядками, послідовності одиниць і нулів. Однак, для послідовностей не бінарних символів немає загального способу для попередньо схвалених режимів, які б дозволили отримання зашифрованих даних того ж формату. Шифрування, що зберігає формат (FPE), призначене для даних, які не обов'язково є двійковими. Зокрема, задається будь-який скінченний набір символів, як і десяткові цифри, так і символьні, метод шифрування зі збереженням формату перетворює дані, відформатовані як послідовність символів таким чином, що зашифровані дані будуть мати той самий формат, включаючи довжину

вихідних даних. Таким чином, наприклад, зашифрований номер соціального страхування за допомогою FPE, буде послідовністю з дев'яти десяткових цифр. FPE полегшує шифрування конфіденційної інформації, а також модернізацію технології шифрування для успадкованих додатків, де звичайний режим шифрування не може бути здійсненим. Наприклад, програми баз даних можуть не підтримувати зміни довжини або формату полів даних. FPE з'явився як корисний криптографічний інструмент, чиї програми включають фінансово-інформаційну безпеку і прозоре шифрування полів у застарілих базах даних. На разі, стандартом є два режими шифрування зі збереженням формату, зазначені в публікації [4], скорочено FF1 і FF3, вони є режимами шифрування на основі мережі Фейстеля. FF1 було надіслано Рогвеєм, Спайсом та Білером (анг. Bellare, Rogaway, Spies)[3] до NIST. FF3 був запропонований Брієром, Стерном та Пейріном (анг. Brier, Peyrin and Stern) до NIST [5]. Зокрема, FF3 є еквівалентним компоненту схеми FF1 з 128-бітовим блоком.

Зашифрований текст з алгоритму шифрування зазвичай дуже відрізняється за виглядом від відповідного відкритого тексту.

Приклад 1.1. Якщо ми шифруємо рядок з відкритим текстом "4111111111111111" за допомогою AES-ECB, ми можемо отримати рядок типу "MKSqwaywf4N8i9gEci4yTUTPalvnQBlBi + Uz6j1Tjig =" для нашого зашифрованого тексту. Якщо оригінальний рядок містить номер кредитної картки, зашифрована версія взагалі не буде виглядати як номер кредитної картки. Вихідна послідовність буде містити символи, відмінні від цифр від {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}, і буде довшим, ніж типовий номер картки 16-символьного відкритого тексту (у цьому прикладі показано кодування Base64).

Зміна формату даних викликає проблеми у багатьох сучасних IT-середовищах, оскільки деякі програми можуть обробляти лише дані, які мають певний формат, а модифікації, які вирішують цю проблему, можуть бути дуже дорогими.

Підхід, який добре працює у багатьох випадках полягає у тому, щоб

адаптувати дані до середовища, замість адаптації середовища до даних і одним зі способів цього, є реалізація шифрування таким чином, що зашифрований текст має той же формат, що і відповідний відкритий текст.

Запровадити алгоритм шифрування до систем — легко, але розробити стійку схему шифрування, відповідно до потреб системи — це вже не є простим завданням.

Щоб отримати зашифрований текст, який має той же формат, що і відповідний відкритий текст, дослідники запропонували багато версій шифрування зі збереженням формату. Технологія датується щонайменше 1981 роком, коли оригінальні рекомендації уряду США, що до впровадження стандарту шифрування даних (DES) (FIPS PUB 74, "Керівництво по впровадженню стандарту шифрування даних NBS квітень 1981 р. [6]), включили опис використання шифрування DES таким чином, що зберегли формат даних: символ за символом — відображення десяткової цифри на іншу десяткову цифру, наприклад.

Протягом наступних років, дослідники запропонували інші спеціальні підходи для використання шифрування зі збереженням формату, але в 2002 році криптографи Блек і Рогвей [7] описали три способи шифрування зі збереженням формату і довели, що запропоновані схеми є стійкими. Режими FFX являють собою еволюцію одного з цих підходів і є значним кроком у застосуванні в криптографії.

«Приватний сектор» вже прийняв технологію шифрування зі збереженням формату. Сьогодні FPE щодня захищає мільярди платежів. Схеми використовуються в процесах охорони здоров'я, в системах обробки авіаліній, у мобільних додатках, у великих базах даних (включаючи Hadoop (проект фонду Apache Software Foundation, вільно розповсюджує набори утиліт, бібліотек і фреймворків для розробки і виконання розповсюджених програм, що працюють на кластерах із сотен і тисяч вузлів)), у веб-транзакціях, а також у наданні організаціям перенесення конфіденційних даних у хмару, зберігаючи повний контроль

над своїми даними.

Модель шифрування зі збереженням формату. Будь-який механізм безпеки такого різновиду шифрування, має бути стійким з певними обмеженнями:

1) Зловмисник знає формат і тип даних у базі даних. Ми повинні припустити, наприклад, що зловмисник шукає номери кредитних карт або номери соціального страхування.

2) Розмір відкритого тексту відносно невеликий, порівняно з типовим розміром блоків даних відомих шифрів. (Блочний шифр типу AES працює на 128-бітових блоках, а номери соціального страхування складають приблизно 28 біт або кредитна картка має довжину в межах 58 біт).

3) Неможливо розширити дані. Коли алгоритм FPE шифрує N-значний номер, він повинен вивести N-значний номер.

4) Дані повинні бути зашифровані детерміновано. Зазвичай, дані шифруються в базі, і дуже бажано зберегти можливість використовувати стовпець як ключ або індекс, що вимагається від множини елементів даних, які шифруються ідентичними елементом даних (при використанні одного ключа).

Зауважимо, що обмеження (4) також можна розглядати як наслідок (3). Жодні додаткові дані (символи) не можуть бути збережені в полі даних, тобто не можна застосовувати випадковість, необхідну для векторів ініціалізації або інші псевдовипадкові генератори.

При цих обмеженнях нам потрібен алгоритм, який шифруватиме дані за допомогою перестановки рядків (символів) в заданому форматі для різних рядків (символів) у тому ж форматі, передбачаючи витік інформації до зловмисника, який має доступ до великої кількості зашифрованих текстів, і може провести перебір розумної кількості пар відкритого/зашифрованого текстів. Алгоритми, які відповідають цьому стандарту, побудовані таким чином, щоб залишатись стійкими в термінах шифрування зі збереженням формату і як правило, не рекомендується

вбудовувати в такі схеми, де можна вважати, що зловмисник має доступ до шифрування/розшифровування або доступ до необмежених пар шифрованого/відкритого текстів.

Можна побудувати стійкий шифр при цих обмеженнях, тому що в той час як зловмисник знає, що шукає конкретний тип даних (наприклад, номер соціального страхування), ми повинні знайти один конкретний номер, який відповідає іншим персональним даним у базі. Тобто, ми можемо переставляти набір чисел і вимірювати успіх зловмисника, на скільки успішним може бути його атака, щоб вгадати певне число, яке надається переставленому номеру та інші визначені дані. Практична мета безпеки полягає в тому, щоб перетворити атаку (крадіжку бази даних через додаток, файл журналу або резервну копію), яка потребує активної роботи злому функції шифрування/розшифрування і збереже дані навіть якщо значну кількість шифрувань та розшифровувань доведеться уникнути.

Шифрування невеликих множин. Зазвичай, для забезпечення конфіденційності та автентичності, у сучасній криптографії використовується шифрування, де розмір повідомлення може бути довільний, крім того, зараз доступні багато методик і конструкцій, які дозволяють замінити чи фіксувати розмір вхідних даних для шифрування. Часто виникає потреба у шифруванні за допомогою симетричного шифру зовсім невеликих повідомлень як то 5-значний поштовий індекс, 4-значний PIN-код або 10-значний ідентифікаційний код, але за умови, що шифротекст також буде мати невеликий розмір (наприклад, з міркувань оптимізації обсягу використовуваної пам'яті). Тобто, без втрати загальності необхідним є застосування симетричного шифру (див. означення 1.2) для невеликого значення натурального числа n (довжина повідомлення). Використання ендоморфного шифру також не обмежує загальності.

Одним з можливих варіантів вирішення цієї задачі є використання моделі блочного шифрування (див. означення 1.2), де довжина

повідомлення не велика, аби використовувати відомі блочні шифри. Якщо використовувати відображення множини повідомлень X у множину шифрованих текстів Y за допомогою шифру AES, використання самого шифру для шифрування, а потім використати обернене відображення результату шифрування у множину повідомлень X, при малій довжині повідомлення, ставить під сумнів стійкість таких конструкцій, у яких потужність множини повідомлень є значно більшою. Зазвичай, блочні шифри (див. означення 1.2) мають довжину блоку 64 або 128 біт.

Приклад 1.2. Розглянемо 1-бітний блочний шифр з 128-бітовим ключем. Очевидно, що 0 буде шифруватись як 0 або 1, для кожного раунду при фіксованому ключі. За допомогою одного відомого вхідного і вихідного біта може бути встановлена повна таблиця пошуку двох записів, тоді система порушена — зловмисник просто шукає кожен біт зашифрованого тексту в таблиці пошуку і ігнорує ключ.

Якщо використовувати AES як раундову функцію для шифрування повідомлень малої довжини, що базується на застосуваннні мережі Фейстеля, то надійність такого шифру стверджувати не можна, коли зловмисник зробить N запитів.

Для роботи з невеликим розміром вхідних даних потрібна нова ідея: необхідно реалізувати випадкове перемішування. Коли довжина вхідних даних N є достатньо малою, можна дозволити приблизно $O(N)$ часу для шифрування або розшифровування даних. Якщо перестановка дійсно випадкова — реалізація таких алгоритмів відповідає принципам шифрування зі збереженням формату.

Випадкові перестановки незамінні і широко поширені в різних додатках. Існує багато алгоритмів для генерації великих випадкових перестановок.

Означення 1.6. Псевдовипадкова функція [8] — це функція, що визначена над (K, X, є ефективно обчислюваною і детермінованою, яка

повертає псевдовипадкову вихідну послідовність Y:

$$F\colon K \times X \to Y$$

Тобто, сімейство псевдовипадкових функцій (PRF) — це сукупність ефективно обчислюваних функцій, що імітують випадковий оракул: не існує дієвого алгоритму, що може розрізнити між функцією випадково обраною з PRF і випадковим оракулом (функція чий вихід зовсім випадковий). Псевдовипадкові функції повертають вихід, який невідмінний від випадкових послідовностей. Вхід може бути довільним, але вихідні дані повинні завжди виглядати абсолютно випадковими.

Означення 1.7. Псевдовипадкова перестановка — це перстановка, визначена над (K, X), яка є ефективно обчислюваною і детермінованою функцією, що повертає псевдовипадкову вихідну послідовність:

$$F\colon K \times X \to X$$

1.$E(K, \cdot)$ — бієктивна функція;

2. існує ефективний алгоритм обчислення оберненої функції $D(K, X)$

Псевдовипадкову перестановку неможливо відрізнити від випадкової перестановки (тобто перестановка, реалізована випадково, незалежно з рівномірною ймовірністю, з сімейства всіх перестановок на області визначення функції). Псевдовипадкові перестановки, які були введені Лубі і Ракофом [9], формалізують криптографічне поняття стійкості блочних шифрів (див. означення 1.2). Так як шифрування кожного блоку відкритого тексту є єдиним блоком тієї ж довжини, секретний ключ можна вважати правилом перестановки для блоків. Перевагою блочних шифрів (у порівнянні з використанням псевдовипадкових функцій для шифрування) є те, що відкритий текст і зашифрований текст мають однакову довжину. Ця властивість зберігає і запобігає втраті пропускної здатності зв'язку. Крім того, це дозволяє легко включати схеми шифрування в існуючі протоколи або апаратні компоненти.

Лубі і Ракоф [9] визначили безпеку псевдовипадкових перестановок за аналогією з різними типами атак (див. означення 1.3, 1.4):

1. Псевдовипадкові перестановки можна інтерпретувати як блочні шифри, які захищені від адаптивних атак на основі обраного шифрованого тексту. Неформально це означає, що криптоаналітик (зловмисник), з доступом до зашифрованих повідомлень за його вибором, не може розрізнити шифр від значення справді випадкової перестановки.

2. Сильні псевдовипадкові перестановки можна інтерпретувати як блочні шифри, які стійкі до атак на основі адаптованого обраного відкритого тексту і атаки на основі обраного зашифрованого тексту.

Хорошим прикладом випадкової перестановки є перемішування карт. Випадкове перемішування карт має багату історію та вплив на різні області математики (комбінаторика, теорія ймовірності, криптографія та інші). Різні підходи перемішування можна використовувати по-різному (наприклад, як схему шифрування, що нас цікавить). Перемішування карт варто виконувати випадковим чином.

Якщо перемішування карт для випадковості використовує PRF (керується елементом випадковості) і можна простежити траєкторію карти не звертаючись до інших карт у колоді — такий спосіб варто розглянути у схемах шифрування даних.

1.3 Види карткових перемішувань

Для подальшої роботи зі схемами перемішування карт ми будемо розглядати простір $X = (x_0, x_1, ..., x_{N-1})$ — послідовність карт. Всього карт — N штук. Для зручності потужність множини карт обирають $|X| = N = 2^n$, $n \in \mathbb{N}$. Також, нам знадобиться елемент випадковості: ми будемо підкидати чесну монету (рівноймовірну, незалежну).

Розглянемо значення підкинутої чесної монети c_i як вихід певної функції C: ймовірність того що $P\{c_i = 0\} = P\{c_i = 1\} = 1/2$, ($\{c_i = 0\}$ — випав «герб», $\{c_i = 0\}$ — випала «решка»).

Гарним способом отримання випадкової перестановки є перемішування карт. Кожен раунд в процедурі перемішування карт є перестановкою на множині X карт (тобто, перестановка на множині $X = \{x_0, x_1, ..., x_{N-1}\}$). У випадку з Елдусом і Дяконісом (анг. Aldous, Diaconis) [10], перемішування карт визначене як перестановка рівномірно обраним $(N/2)$-бітовим рядком.

Для отримання випадкової перестановки, можна побудувати m-арну δ-залежну перестановку — узагальнити алгебраїчну конструкцію попарно незалежних перестановок. У роботі Ноара і Рейнхольда [11] запропонована 3-транзитивна рефлексивна група перестановок.

Означення 1.8. Група перестановок G над множиною X є рефлексивною k-транзитивною, якщо для перестановки над групою $X = \{ x_0, x_1, ..., x_{N-1} \}$, що є підгрупою симетричної групи S_n для кожних двох m-кортежів $\{a_1, ..., a_k\}$ та $\{p_1, ..., p_k\}$ окремих елементів з X визначена перестановка $\xi \in G$: для кожного i в межах $1 \leq i \leq k$, $\xi(a_i) = p_i$.

m-транзитивна рефлексивна група перестановок m-арно незалежна і є алгебраїчною конструкцією з попарно незалежних перестановок, що визначені рефлексивними 2-транзитивними групами перестановок (містять всі лінійні перестановки). Ноаром і Рейнохольдом запропоновано [11] використовувати відомі конструкції рефлексивної 3-транзитивної групи перестановок. Однак цей підхід не може бути узагальнений для більших значень m: з класифікації числа простих скінченних груп випливає, що для певного значення m не існує рефлексивних m-транзитивних груп над X. симетричні групи S_n і змінна група A_n — є лише декілька таких груп для $m = 4$ та $m = 5$. Треба бути обережним, щоб не тлумачити так, що для $m = 4$ не існує ефективних алгебраїчних конструкцій m-арних перестановок. Однак, потрібно зауважити, що при

$m = 4$ будь-яке мале сімейство m-арних перестановок — не є групою перестановок.

Перемішування карт Riffle — це один з найперших алгоритмів, що виник дуже давно і почав удосконалюватись Кнутом [12] ще у далекому 1949. З того часу було створено багато способів перемішування карт, одні були кращі, інші — гірші. Для отримання результатів, потрібно ознайомитись з такими перемішуваннями на практиці. Найкращий спосіб це зробити,— переглянути чим керуються розглянені у цьому розділі способи перемішування.

Означення 1.9. Перемішування карт — це процедура, що використовується для отримання випадкової послідовності на множині гральних карт, щоб забезпечити елемент випадковості в карткових іграх.

Перемішування карт використовуються в прикладних задачах математики, в часності криптографії. Перестановка Торпа делатьно розглянена у роботі Бена Моріса, Філіпа Рогвея [13] для шифрування зі збереженням формату, але використання і аналіз шифру були запатентовані за довго до цього. Найбільше робіт у Бена Моріса [13], [14], [15], [16], [17]. Торп став одним з найпопулярніших з усіх карткових перенесень на схеми шифрування в криптографії. Слід розглянути як саме потрібно перемішувати за допомогою алгоритму Торпа.

Перемішування Riffle Поширена технологія перемішування карт — «переплетення», множина X ділиться навпіл і карти в новій послідовності підлягають розподілу (див. рис. 1.1):

(a) Почнемо з впорядкованої множини карт.

(b) Ділимо множину на дві частини.

(c) Дві частини «переплітаються» разом.

(d) Дві частини утворюють «переплетену» послідовність.

Точну математичну модель «переплетеного» перемішування було введено Гільбертом і Шеноном [18]. Однак, таку конструкцію важко сприйняти як щось «випадкове». Тому, розглянемо більш складну модель

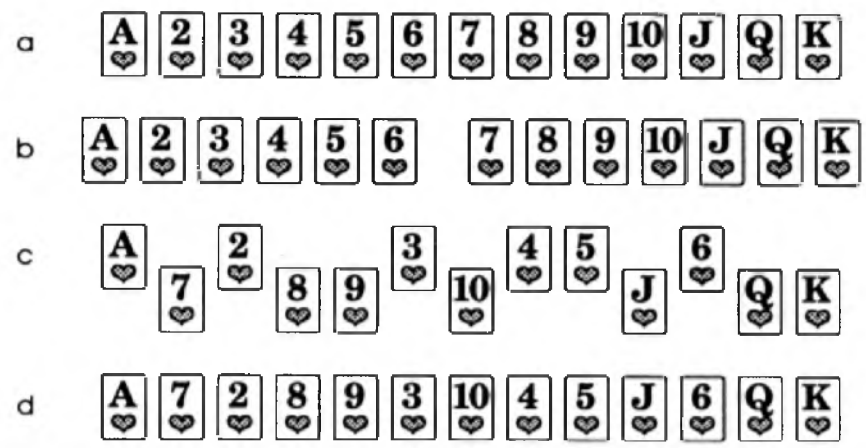

Рисунок 1.1 – Перемішування карт Riffle

Riffle, запропоновану Дональдом Кнутом [12], яка використовує генератор псевдовипадкових чисел.

Перемішування послідовності карт $\{x_0, x_1, x_2, x_3, ..., x_{N-1}\}$ виконується наступним чином: генерується послідовність випадкових чисел з множини позицій карт $\{0, 1, 2, ..., N-1\}$: $S^{rand} = \{s_0, s_1, ..., s_{N-1}\}$. Виконуючи прохід по множині карт $\{x_0, x_1, x_2, x_3, ..., x_{N-1}\}$, кожну карту x_i міняємо місцями з картою на позиції s_i.

Приклад 1.3. Перемішування Riffle. Нехай задано послідовність $\{x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7\}$, $N = 8$. Тоді, якщо згенеровано послідовність псевдовипадкових чисел: $S^{rand} = \{7, 4, 2, 5, 5, 0, 1, 6\}$, послідовність перемішаних карт буде мати вигляд: $\{x_0, x_5, x_3, x_2, x_1, x_4, x_7, x_6\}$.

Правило перемішування $E_K^{Riffle}(X) = Y$. Виконуємо прохід по масиву $X = (x_0, x_1, ..., x_{N-1})$. Кожний x_j міняємо місцями з x_{k_j}. Продовжуємо для всіх позицій масиву карт $j \in \{0, 1, ..., N-1\}$.

Отже, випадковістю при перемішуванні у схемі буде $S^{rand} = \{s_0, s_1, ..., s_{N-1}\}$ — послідовність випадкових чисел з множини індексів позицій карт. Отримане перемішування залежить від значень кожного (s_i).

Алгоритм класичного перемішування Riffle.

Вхідні дані: N, X, r

Результат: X

цикл $i := 0$ *до* $r - 1$ **виконати:**

  цикл $j :=$ *0 до N-1* **виконати:**

    $a := X$[i];

    $X[i] := X[N - 1 - i]$;

    $X[i] := a$;

  кінець

кінець

Алгоритм 1: Псевдо-код класичного алгоритму перемішування Riffle.

Алгоритм перемішування Riffle з генератором псевдовипадкових чисел.

Вхідні дані: N, X, r, K

Результат: X

цикл $i := 0$ *до* $r - 1$ **виконати:**

  цикл $j :=$ *0 до N-1* **виконати:**

    $a := X$[i];

    $X[i] := X[K[i]]$;

    $X[K[i]] := a$;

  кінець

кінець

Алгоритм 2: Псевдо-код алгоритму перемішування Riffle з генератором псевдовипадкових чисел.

На разі, перемішування Riifle реалізовано в багатьох бібліотеках мов програмування як алгоритм отримання випадкової послідовності масиву даних. Дане перемішування являється простим та швидким.

Перемішування Торпа. Послідовність карт ділиться навпіл і для кожного раунду перемішування необхідно підкинути чесну монету. Для кожного проходу перемішування необхідно розглянути пари карт x_i та $x_{i+N/2}$, $i \in \{0, 1, ..., N/2 - 1\}$. Для кожної пари карт x_i та $x_{i+N/2}$ на позиціях i та $i + N/2$ застосовуємо функцію C (підкидуємо чесну монету). Якщо вихід функції C: $c_i = 1$ — пару на позиціях i та $i + N/2$ ставимо на місця $2i$ та $2i + 1$, інакше, якщо $c_i = 0$ — пару x_i та $x_{i+N/2}$ ставимо на місця карт з позиціями $2i + 1$ та $2i$.

Приклад 1.4. Перемішування Торпа. Нехай задано послідовність $\{x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7\}$, $N = 8$. Тоді після послідовного підкидання монети, події якої зафіксовані наступним чином: $\{c_1 = 1, c_2 = 1, c_3 = 0, c_4 = 1\}$, перемішана послідовність карт буде мати вигляд: $\{x_4, x_0, x_5, x_1, x_2, x_6, x_7, x_3\}$.

Перемішування Swap-Or-Not. Конструкція Swap-Or-Not була винайдена, як спосіб перемішування колоди карт [17].

Розглянемо такий самий набір карт $X = (x_0, x_1, ..., x_{N-1})$. Нехай для кожного кроку i, $i \in \{0, 1, ..., N - 1\}$ значення підкинутої монети буде рівним c_i.

Кожну пару карт x_i та $x_{i\oplus c_i}$, на позиціях i та $i \oplus c_i$ переставляємо місцями, якщо $\{c_i = 1\}$, інакше залишаємо на місці.

Приклад 1.5. Перемішування Swap-Or-Not. Нехай задано послідовність карт $\{x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7\}$, $N = 8$. Тоді після процедури підкидання чесної монети, значення якої $\{c_0 = 1, c_1 = 0, c_2 = 0, c_3 = 1, c_4 = 1, c_5 = 1, c_6 = 0, c_7 = 1\}$ послідовність перемішаної множини карт буде мати вигляд: $\{x_7, x_4, x_2, x_1, x_0, x_3, x_6, x_5\}$.

Перперемішування Фішера-Єтса

Один з найпростіших і відомий у своєму широкому застосуванні

алгоритм (на основі заданої послідовності різних карт $\{x_0, x_1, ..., x_{N-1}\}$) для отримання випадкового перемішування Фішера-Єетса або Кнута [12].

Перемішування множини X з випадково обраним елементом з $\{x_0, x_1, ..., x_{N-1}\}$ (кожен з яких може бути обраний з однаковою ймовірністю), потім дана процедура повторюється для $\{x_{N-2}, x_{N-3}, ..., x_1\}$ [19]. Такий алгоритм виявився не оптимальним у ситуаціях, коли необхідnt часткове перемішування множини [7], при реалізації на структурі даних, таких як список. Пряме генерування рівномірної випадкової величини призводить до зниження швидкості, роблячи його менш ефективним, ніж коли N дуже велике [19]; кількісь затраченої пам'яті робить його не ефективним в паралельній реалізації. Зауважимо, що ефективність цього алгоритму залежить від наявності незалежного та рівноймовірного генератора випадкових чисел.

Алгоритм RS: Divide-And-Coin

Алгоритм RS описаний [19] тільки в двійковому вигляді, якщо наявна чесна монета. Розглянемо алгоритм у прикладі.

Приклад 1.6. Для послідовності $\{x_0, x_1, x_2, x_3, x_4, x_5\}$ генеруємо випадкову послідовність біт: $\{1, 0, 1, 1, 0, 0\}$: 0-група це $\{x_1, x_4, x_5\}$, 1-група це $\{x_0, x_2, x_3\}$, які можна записати у наступному вигляді: $\{1, 0, 1, 1, 0, 0\}$, $\{x_0, x_2, x_3\}$. Для 0-групи та 1-групи знову генеруємо випадкові послідовності, наприклад $\{0, 0, 1\}$, $\{0, 1, 0\}$ відповідно. Потім поєднуємо $\{x_1, x_4\}$, $\{x_5\}$, $\{x_0, x_3\}$, $\{x_1\}$, якщо для пар випали біти $\{0, 1,$ отже отримали випадкову перестановку: $\{x_1, x_4, x_5, x_3, x_0, x_2\}$

Ресел (анг. Ressel) [20] запропонував модифікацію для RS: Divide-And-Coin алгоритму, для випадкового перемішування структури або списку, використовуючи подібну ідею «Divide-And-Conque» («Розділяй та володарюй»), але таку схему перемішування реалізувати набагато складніше.

Для практичної реалізації більш ефективним є застосування цього

алгоритму детермінованим чином без рекурсій за стандартом визначеним в [21].

Алгоритм RS(N, X)

Вхідні дані: N, X, r

Результат: X

Крок 1:

Визначимо дві множини: N_0, N_1;

цикл *i := 0 до N-1* **виконати:**

обираємо випадковий біт C[i] з множини {0, 1}

якщо *C[i] = 0* **тоді:**

X[i] помістимо до множини N_0;

кінець

інакше

X[i] помістимо до множини N_1;

кінець

кінець

Крок 2:

цикл *Для кожного k* **виконати:**

якщо $k = 1$ **тоді:**

зупинити роботу алгоритму;

кінець

якщо $k = 2$ **тоді:**

обираємо випадковий біт b з множини {0, 1};

якщо *b = 1* **тоді:**

змінюємо на зворотній порядок для групи;

кінець

кінець

якщо $k > 2$ **тоді:**

повторити кроки 1-2;

кінець

кінець

Алгоритм 3: Псевдо-код алгоритму RS: Divide-and-coin.

Ідеальне перемішування. У роботі Маргонди (анг. Margonda) [22] запропоновано алгоритм ідеального перемішування (PSCA). Описане перемішування має багато цікавих математичних властивостей і застосувань в інформатиці [23].

Для перемішування відкритого тексту $(x_0, x_1, ..., x_{N-1})$, потрібно використовувати натуральне число k, $k \leq N$. Перемішування працює наступним чином: По-перше, це виконання перемішування на $(x_1, x_2, ..., x_{a^n})$, де $N = 2^n$, потім виконується обмін.

Висновки до розділу 1

У даному розділі було розглянено матеріали за тематикою дослідження. Шифрування зі збереженням формату, являє собою комплекс заходів які забезпечують конфіденційність, цілісність даних, при цьому, таке шифрування може зберегти формат відкритого тексту у зашифрованому. Показано, чому варто використовувати схеми та алгоритми, які зберігають довжину повідомлення або формат.

У розділі визначені відомі карткові перемішування. Такі перемішування, в свою чергу, можна використовувати для схем шифрування зі збереженням формату, в часності, для шифрування повідомлення невеликої довжини.

Визначений термін перемішування, та розглянено, чому варто використовувати випадкове перемішування карт для отримання псевдовипадкової перестановки.

2 ШИФРИ, ЩО БАЗУЮТЬСЯ НА СХЕМАХ ПЕРЕМІШУВАННЯ КАРТ

У даному розділі запропоновані шифри, що використовують методи перемішування карт. Перемішування карт можна розглянути як схеми шифрування, деякі методи перемішування карт з попереднього розділу перенесено на схеми шифрування. Показано властивості таких шифрів та алгоритми їх побудови. Проаналізовано та показано, чому їх варто використовувати у питаннях шифрування зі збереженням формату на невеликій множині.

2.1 Отримання шифрів зі схем перемішування карт

Перемішування карт можна розглянути як схему шифрування, і навпаки. Якщо існує метод перемішування N карт, це визначає відповідний спосіб шифрування множини карт X.

Якщо перемішати множину карт і потім переглянути нові позиції, на які перемістились карти — це подібно виклику раундової функції в блочному шифрі. Випадковість, що використовується при перемішуванні відповідає секретному ключу схеми шифрування.

Розглянемо деталніше карткові перетворення на множині карт Σ_N. Всього існує N! різних перестановок (різних можливих послідовностей). Якщо орієнтуватись на блочний шифр AES, розмір блоку якого 128 біт, усього різних можливих блоків – 2^{128}. Для отримання такого ж числа різних можливих шифротекстів, нам знадобиться більше ніж 34 карти.

$$N! \approx \sqrt{2\pi N} \cdot \left(\frac{N}{e}\right)^N$$

(Формула Стірлінга асимптотичного обчислення факторіалу)

Тоді починаючи з кількості 35 карт, можна отримати більше ніж 2^{128} різних можливих послідовностей:

$$35! > 2^{128}.$$

Для гарного карткового перемішування потрібна ідея простого (примітивного, анг. oblivius) перемішування, яку запропонув Наор [11].

Твердження 2.1. *Просте перемішування (англ. oblivious shuffle) — це процедура перемішування карт, що вимагає лише невеликої кількості раундів і забуває розташування карти після кожного раунду.*

Тобто, можна простежити траекторію карти, не звертаючись до інших карт у колоді.

Розглянемо більш детально значення простого перемішування (див. означення 2.1). Кожну і-ту карту можна розглядати як відкритий текст x_i, $X = \{x_0, ..., x_{N-1}\}$, $i \in \{0, 1, ..., N-1\}$. Тобто, при шифруванні методами, що базуються на простому перемішуванні, якщо нам потрібно перемішати певний відкритий текст x_i, не потрібно перемішувати всю послідовність $X = \{x_0, ..., x_{N-1}\}$, потрібно лише простежити, який відкритий текст x_j переходить на відповідну позицію відкритого тексту x_i.

Узагальнюючи, при простому перемішуванні N карт для визначення положення однієї карти x_i необхідно не більше $\mathcal{O}(\log N)$ біт.

Під час шифрування слід використовувати не тільки перемішування, а і раундові перетворення вхідних даних. Для нашої задачі шифрування зі збереженням формату як найкраще підходить використання шифрів, що базуються на методах перемішування карт.

Застосування випадкового карткового перемішування у криптографії було помічено багато років тому. Наприклад, Ноар і Реінхольд (анг. Naor, Reingold) [11] для схеми шифрування застосовували перемішування Торпа. Процеси перемішування карт, очевидно, подібні блочним шифрам, тому різні алгоритми, що базуються на перемішуванні карт часто використовуються як блочні шифри у криптографічних схемах [24]. З

огляду на причини ефективності тільки прості карткові перемішування (див. означення 2.1) є хорошими кандидатами в блочні шифри.

Перемішування карт (див. розділ 1.3) можна розглянути як схеми шифрування. Для цього, кожну карту x_i слід розглянути як відкритий текст (наприклад, як і-тий біт вхідного повідомлення).

Стійкість криптографічної схеми, яка використовує певну карту у схемі шифрування як блок, залежить від якості перемішування, взятого в основу для шифру. Це можна визначити тим, як дане перемішування карт наближається до рівномірного розподілу [25]:

1. швидкість збіжності до стаціонарного розподілу (залежить від кроків самого алгоритму);

2. кількість кроків, виконаних алгоритмом (зокрема, кількість кроків, необхідних для того, щоб розподіл перемішаної послідовності на цей крок був близький до рівномірного розподілу).

Розглянемо способи шифрування, що базуються на схемах перемішування карт, які було розглянуто у розділі (див. розділ 1.3) та порівняємо їх властивості: чому варто використовувати саме такі схеми для шифрування зі збереженням формату.

Шифрування перемішуванням Торпа [13]. Більше 20 років тому Ноар [11] довів, що перемішування Торпа є простим перемішуванням (див. твердження 2.1). Якщо перемішування Торпа при перемішуванні досягає випадкової послідовності карт досить швидко, ця властивість робить його придатним для шифрування, зокрема для шифрування зі збереженням формату на малих множинах.

Для вхідного повідомлення X та кожного раунду $i \in \{0, 1, ..., R-1\}$ ключем $K_i = (k_{i0}, k_{i1}, ..., k_{i(\frac{N}{2}-1)})$, де K_i — раундовий ключ і-того раунду шифрування, k_{ij} — j-тий біт раундового ключа K_i. Послідовність бітів $k_{ij}, j \in \{0, 1, ..., \frac{N}{2} - 1\}$ утворюють випадкову, рівноймовірну та незалежну послідовність виходів функції підкидання чесної монети: $P(k_{ij} = 0) = 1/2 = P(k_{ij} = 1)$.

Правило перемішування для $E_k^{Thorp}(X) = Y$ [13]. Послідовність X, для раунду i, ділиться на дві рівні частини: по $\frac{N}{2}$ кожна. Далі обирають з лівої та правої частини x_j та $x_{\frac{N}{2}+j}$ та виконують перемішування за допомогою раундового ключа K_i, в залежності від значення k_{ij} для раунду i. Якщо $k_{ij} = 0$ переставляємо місцями з x_{2j} та x_{2j+1}, інакше перестявляємо місцями з x_{2j+1} та x_{2j} відповідно.

Узагальнюючи, потрібно використати ключ довжиною $R \cdot \frac{N}{2}$ випадкових біт, який потрібен для перемішування вхідних даних. Знадобиться лише R викликів PRF.

Практична реалізація шифру на основі алгоритму випадкового перемішування Торпа. $E_{K,R}^{Thorp}(X)$ — алгоритм шифрування Торпа для вхідних даних X з ключем K та кількістю раундів R.

Більш ефективною реалізацією $E_{K,R}^{Thorp}(X)$ є техніка, запропонована Беном Морісом, Філіпом Рогвеєм та Тілом Стейджерсом [26], яка дозволить одним викликом забезпечити п'ять раундів шифрування або розшифровування (див. рис. 2.1).

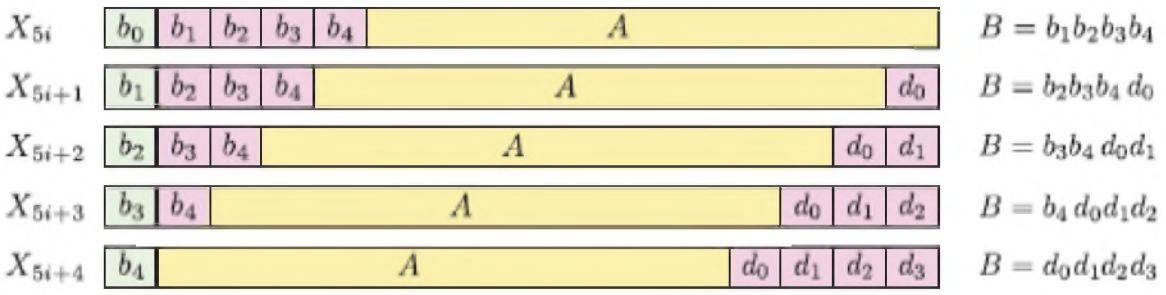

Рисунок 2.1 – Приклад реалізації детермінованого шифрування алгоритмом Торпа з модифікацією п'ятикратного прискорення.

Зауваження. Шифрування алгоритмом з модифікацією п'ятикратного прискорення Торпа виконує шифрування блоків даних $\{0, 1\}^{128} \to \{0, 1\}^{128}$.

Реалізація $E_{K,R}^{Thorp}(X)$ з модифікацією п'ятикратного прискорення для R раундів ефективно використовуватиме інше правило (ніж наведено раніше для звичайного перемішування), коли $N = 2^n$ і $C(x, r) = p(x \bmod \frac{N}{2}, r)$, що відповідає незбалансованій мережі Фейстеля, де $p = f(K, \cdot)$— вихід з PRF (див. означення 1.6), f: $K \times \Sigma^* \to \{0, 1\}^n$.

Послідовні n-бітові рядки x_j (Рисунок 2.1) шифруємо піднімаючись вгору або розшифровуємо, спускаючись вниз. Для будь якого $A \in \{0, 1\}^{n-5}$ та раунду j, такого що j ділиться на 5, один виклик обчислює функцію C для усіх $(n-1)$-бітових рядків:

A для раунду j: * * * * A,
A для раунду j + 1: * * * A *,
A для раунду j + 2: * * A * *,
A для раунду j + 3: * A * * *,
A для раунду j + 4: A * * * *,
(*) – може бути 0 або 1.

Якщо $v_1, ..., v_k$ — k-бітний рядок або ціле число, тоді при використанні шифру $E_{K,R}^{Thorp}(X)$, $(v_1, ..., v_k)$ – це кортеж, зашифрований деяким фіксованим чином.

Позначимо зашифрований текст $Y \in \{0, 1\}^n$ після i раундів шифрування відкритого тексту X схемою $E_{K,R}^{Thorp}(X)$. Замість того, щоб оцінювати p, витягнемо достатню кількість бітів, щоб визначити усі $C(U, r)$, які можуть бути необхідні в тій же групі із п'яти послідовних раундів.

Реалізація цієї ідеї дещо складна, оскільки важливо щоб кожен вихід псевдовипадкової функції p (послідовність значень результатів подій підкидання чесної монети $C(U, r)$) був чітко визначений і водночас кожен вихід був незалежним з $C(V, s)$, $C(V, s) = C(U, s)$.

Стратегія проілюстрована на рисунку (див. рис. 2.1). Використаємо той факт, що для $j \in \{0, 1, 2, 3, 4\}$ рядки X_{5i}, X_{5i+j} мають $(n-5)$-бітових спільних рядків з A. Оцінюємо p тільки на $< A, i >$, використовуючи

$< B, j >$, де B – це конкатенація бітів 1, ..., $(4-j)$ та $(n-j+4)$, ..., $n-1)$ з рядка $X_{5i,j}$.

Узагальнюючи даний підхід, використання перемішування Торпа відбувається шляхом заміни бінарних операцій арифметикою за модулем, що дає п'ятикратний приріст швидкості шифру $E_{K,R}^{Thorp}(X)$ для довільної потужності вхідних даних N, за умови, коли N кратна 32. П'ятикратне прискорення використовує $5 \cdot 2^4 = 80$ зі 128 бітів виходу псевдовипадкової функції (PRF) для кожного раунду.

Можна втілити довільне k-кратне прискорення, якщо PRF-вихід буде дорівнювати $k \cdot 2^{k-1}$ бітів, хоча це вимагає округлення n до наступного числа, що кратне 2k.

Алгоритми шифрування та розшифровування перемішуванням Торпа з модифікацією п'ятикратного прискорення використовують функцію $F_p^r(x)$.

Алгоритм $F_p^r(x)$

Вхідні дані: N, x, r

Результат: c

$i := r\text{div}5;$

$j := r\text{mod}5;$

$a := (x\text{div}2^j)\text{mod}N/32;$

$hi := (x\text{div}2^j)\text{div}N/32;$

$lo := x\text{mod}2^j;$

$b := hi \cdot 2^j + lo;$

$b := hi \cdot 2^j + lo;$

$Y := (N,i,a);$

$table := p(Y);$

$k := 16j + b;$

$c := table\,[k]);$

Алгоритм 4: Псевдокод алгоритму F_p для перемішування Торпа.

Алгоритм $E_{K,R}^{Thorp}(X)$:

Вхідні дані: N, X, r

Результат: X

цикл *0 до* $R-1$ **виконати:**

присвоїти значенню c вихід раундової функції $F_p^r(X \bmod N/2)$;

якщо $X < N/2$ **тоді:**

$X := 2 \cdot X + c$;

кінець

інакше

$X := 2 \cdot (X \bmod N/2) + 1 - c$;

кінець

кінець

Алгоритм 5: Псевдокод алгоритму $E_{K,R}^{Thorp}(X)$ для перемішування Торпа з модифікацією п'ятикратного прискорення.

Алгоритм $D_{K,R}^{Thorp}(y)$:

Вхідні дані: N, Y, r

Результат: Y

цикл $R-1$ *до 0* **виконати:**

присвоїти занченню c вихід раундової функції $F_p^r(Y \operatorname{div} 2)$;

якщо $c = Y mod 2$ **тоді:**

$Y := Y \operatorname{div} 2$;

кінець

інакше

$Y := Y \operatorname{div} 2 + N/2$;

кінець

кінець

Алгоритм 6: Псевдокод алгоритму $D_{K,R}^{Thorp}(Y)$ для перемішування Торпа з модифікацією п'ятикратного прискорення.

Застосування такого алгоритму на приктиці можливе завдяки роботі Моріса, Рогвея та Стейджерса [13] і доведенню наступної теореми.

Теорема 2.1. *Нехай N ділить 32 і R >= 1*

Якщо p: Σ^* $\rightarrow$ $\{0, 1\}^{128}$ — випадкова функція, виходи якої рівноймовірні та незалежні (тобто h з PRF), тоді перемішування Enc_p — це реалізація E_K^{Thorp} для R раундів. Dec_p — обернена функція до E_K^{Thorp}.

Шифрування перемішуванням Swap-Or-Not [17]. Розглянемо той самий простір повідомлення $X = (x_0, x_1, ..., x_{n-1})$. Для кожного раунду $i \in \{0, 1, ..., r-1\}$ ключем буде випадкова послідовність бітів (виходів функції підкидання чесної монети) $K_i = (k_{i0}, k_{i1}, ..., k_{i(n-1)})$.

Для шифрування нам знадобиться K_i — раундовий ключ блочного шифрування KF, послідовність K_0, ..., $K_{r-1} \in \{0, 1\}^n$ — послідовність раундових ключів, що утворюють ключ K. F_1, ..., F_{r-1} — набір булевих функцій, кожна з яких відображає n-бітовий рядок в 1 біт: F_i: $\{0, 1\}^n \rightarrow \{0, 1\}$.

Перемішування відбувається просто: X приймає занчення $K_i \oplus X$ і раундова функція визначає набір $\{X, K_i \oplus X\}$. Розшифрування Swap-Or-Not ідентичне шифруванню: виконуємо алгоритм від $(r-1)$-кроку до першого. Розглянемо алгоритм більш детально. Для кожного раунду i утворюємо пару значень: $X \in \{0, 1\}^n$ та його «пару» $X' = K_i \oplus X$. Далі пару міняємо місцями або залишаємо, відповідно значенню раундової функції F_i для набору $\{X, X'\}$. Відповідно, щоб F_i була значимою, необхідно обрати $\hat{X} = \max(X, X')$ та помножити на F_i.

Зауважимо, що кожен відкритий текст перетворюється у зашифрований шляхом операції $\oplus$ на підмножину раундових ключів $\{K_0, ..., K_{r-1}\}$. Це виглядає як лінійне перетворення, але не є таким. Кожен відкритий текст відображається в шифрований шляхом операції XOR з раундовими ключами K_0, ..., K_{r-1} та використанням функцій F_0,

..., F_{r-1}, F_i: $\{0, 1\}^n \to \{0, 1\}$.

Алгоритм шифрування Swap-Or-Not. $E^{Swap}_{K,F}(X) = Y$ [17]

Вхідні дані: N, X, r

Результат: X

цикл *i := 0 до r-1* **виконати:**

  цикл *j := 0 до N-1* **виконати:**

    обмраємо випадковий біт $K_i[j]$ з множини $\{0, 1\}$;

    $X'[j] := K_i[j] \oplus X[j]$;

    $\hat{X}[j] := \max(X[j], X'[j])$;

  кінець

  якщо $F_i(\hat{X}) = 1$ **тоді:**

    $X := X'$;

  кінець

кінець

Алгоритм 7: Псевдокод алгоритму Swap-Or-Not.

Шифрування Swap-Or-Not (блочне змішування) [17]. Swap-Or-Not може також розглядатися як процес створення блочного змішування. Для цього розглянемо раундову функцію F_i: $\{0, 1\}^n \to \{0, 1\}$ як швидку блочну конструкцію, що залежить від бітового рядка L_i. Нехай $L_i(\hat{X}) = L_i \bigodot \hat{X}$:

$$L_i \bigodot \hat{X} = L_i[0]\,\hat{X}[0] \oplus L_i[1]\,\hat{X}[1] \oplus \ \ldots \ \oplus L_i[n-1]\,\hat{X}[n-1].$$

$L_i(\hat{X})$ — ефективно обичлювальна функція, що отримується з L_i та $\hat{X}$.

Узагальнення. Можна розлянути шифр Swap-Or-Not в абелевій групі $G = (\Sigma_N, +)$ замість групи $(\{0, 1\}^n, \oplus)$ бітових рядків над операцією побітового додавання (для зручності елементи групи опишемо як $\Sigma_N = \{0, 1, ..., N-1\}$) [17].

Алгоритм шифрування Swap-Or-Not (блочне змішування) $E_{K,F}^{Swap}(X) = Y$ [17].

Вхідні дані: N, X, r

Результат: X

цикл *i := 0 до r-1* **виконати:**

  цикл *j := 0 до N-1* **виконати:**

    обмраємо випадковий біт $K_i[j]$ з множини $\{0, 1\}$;

    $X'[j] := K_i[j] \oplus X[j]$;

    $\hat{X}[j] := \max(X[j], X'[j])$;

  кінець

  якщо $L_i \bigodot \hat{X} = 1$ **тоді:**

    $X := X'$;

  кінець

кінець

Алгоритм 8: Псевдокод алгоритму Swap-Or-Not (блочне змішування).

Згадаємо, що простір повідомлення $|X| = \Sigma_N$ — це степінь двійки, тоді можна зашифрувати повідомлення просто визначивши групу G, скажімо з операцією додавання за модулем N.

Узагальнений алгоритм Swap-Or-Not використовує операцію додавання за модулем N. Послідовність значень K_i — рівномірна і незалежна вибірка з Σ_N. KF — це послідовність раундових ключів K_0, K_1, ..., $K_{r-1} \in \Sigma_N$. Раундові функції також визначені на Σ_N: F_0, F_1, ..., F_{r-1}: $\Sigma_N \to \{0, 1\}$.

Узагальнений алгоритм шифрування Swap-Or-Not. $E_{KF}^{Swap}(X)$ [17]

Вхідні дані: N, X, r

Результат: X

цикл *i := 0 до r-1* **виконати:**

  цикл *j := 0 до N-1* **виконати:**

    $K_i[j] \overset{\$}{:=} \{0, 1\}$;

    $X'[j] := K_i[j] \oplus X[j]$;

    $\hat{X}[j] := \max(X[j], X'[j])$;

  кінець

  якщо $F_i(\hat{X}) = 1$ **тоді:**

    $X := X'$;

  кінець

кінець

Алгоритм 9: Псевдокод узагальненого алгоритму Swap-Or-Not.

Шифрування ідеальним перемішуванням. Алгорритм представлений у роботі Маргонди.

Для шифрування відкритого тексту $(x_0, x_1, ..., x_{N-1})$, необхідно використати ключ — натуральне число k, $k \leq N$.

Алгоритм ідеального перемішування.

Вхідні дані: відкритий текст X, $N \geq 1$, k.

Результат: Кожен елемент шифрованого тексту Y[i] містить результат операції перестановки на P[i] при k раундах.

$n := \log_2(N)$

якщо $n \neq \lceil \log_2(N) \rceil$ **тоді:**

  цикл $i := N$ *до* $2^n - 1$ **виконати:**

    генеруємо випадковий біт x_i з множини $\{0, 1\}$

  кінець

кінець

цикл $i := 0$ *до* k **виконати:**
| **цикл** $j := 0$ *до* $2^n - 1$ **виконати:**
| | **якщо** $(j \mod 2 = 0)$ *і* $(j \neq 0$ *або* $j \neq 2^{n-1})$ **тоді:**
| | | B[j] := T[j/2];
| | **кінець**
| | **інакше**
| | | **якщо** $(j \mod 2 \neq 0)$ *і* $(j \neq 0$ *або* $j \neq 2^{n-1})$ **тоді:**
| | | | B[j] := T[$\lceil j/2 + (2^n - 1)/2 \rceil$];
| | | **кінець**
| | **кінець**
| | **цикл** $j := 0$ *до* $2^n - 1$ **виконати:**
| | | **якщо** $j \mod 2 = 0$ **тоді:**
| | | | Y[j] := B[j+1];
| | | **кінець**
| | | **інакше**
| | | | Y[j] := B[j-1];
| | | **кінець**
| | **кінець**
| | **цикл** $j := 0$ *до* $2^n - 1$ **виконати:**
| | | Y[j] := T[j];
| | **кінець**
| **кінець**
кінець

Алгоритм 10: Псевдо-код алгоритму ідеального перемішування.

Криптоаналітик повинен вміти обчислювати обернену функцію до перемішування, для отримання вхідного повідомлення. Шифрований текст - це послідовність, яка побудована за допомогою операції перемішування та обміну k разів. Для криптоаналізу PSCA [22], використовували аналіз атак обраним відкритим текстом.

2.2 Аналіз шифрів, які базуються на схемах перемішування карт

Перемішування карт має багату історію з математики, починаючи з роботи Маркова і Пуанкаре. Класичною проблемою при перемішуванні карт є кількість раундів перемішувань, тобто кількість, яка необхідна для того, щоб колода карт була гарно змішана. Для більшості відомих алгоритмів перемішування вже отримано оцінки нижніх границь кількості перемішувань (раундів), оцінки стійкості, про що і буде йти мова у цьому підрозділі.

У задачі шифрування, що зберігає формат, потрібно створити шифр на довільному наборі X для деякої розмірності N. Зазвичай використовуються конструкції, які подібні звичайним блочним шифрам, наприклад AES. Дана задача розглянена в багатьох роботах [2], [26], [17], [3], [27], [28], [16], [15], [29], [14], [13] і є предметом поточної роботи стандартизації NIST.

Коли N є достатньо малим, можна дозволити близько $\Omega(N)$ часу для шифрування. Доведено, що це досягається при простому картковому перемішуванні (див. твердження 2.1). А коли N є достатньо великим, що немає зловмисника, який зміг би зробити приблизно $N^{1/2}$ запитів атак,— краще використовувати стандартні криптографічні конструкції, які базуються на застосуванні мережі Фейстеля.

Для нашої задачі (шифрування повідомлень середнього та малого розміру, близько 2^{30} біт), використання відомих блочних шифрів недоречне, так як відомі атаки, що витрачають час, пропорційний розміру вхідного повідомлення. Навіть при збільшенні довжини ключів конструкції, побудовані на збалансованій мережі Фейстеля, не є розв'язком задачі шифрування зі збереженням формату при невеликих вхідних повідомленнях.

Проблема зі збалансованою мережею Фейстеля. Наор і Рейнхольд [11] аналізували незбалансовану мережу Фейстеля (див. рис. 2.2), показавши, що за один прохід максимально незбалансована мережа Фейстеля (яка працює на n бітах), залишається захищеною майже до $2^{\frac{n}{2}}$ атак на основі обраного відкритого тексту.

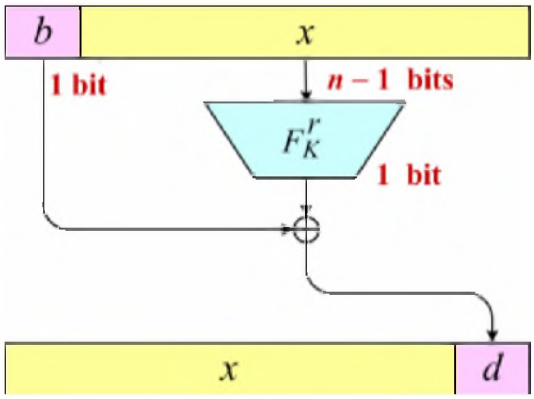

Рисунок 2.2 – Мережа Фейстеля

Для збалансованої мережі Фейстеля Лубі і Ракоф [9] показали, що три раунди забезпечують стійкість до атак на основі обраного відкритого тексту (або стійкість схеми з чотирьма раундами для атак на основі обраного шифрованого тексту) — майже до $2^{\frac{n}{4}}$ запитів атак.

Покращили результати Лубі і Ракофа вчені Мауер і Піетжак (анг. Maurer Pietrzak) [30]. Вони показали, що r раундів збалансованої мережі Фейстеля забезпечують стійкість близько $2^{\frac{n}{2}-\frac{1}{r}}$ запитів атак на основі обраного шифрованого тексту. У своїй роботі Патарін (анг. Patarin) [31] довів, що постійне число раундів (шість для стійкості до атак на основі відкритого тексту) вже вистачає, аби забезпечити стійкість близько до $2^{\frac{n}{2}}$ атак. Також він припустив, що достатня кількість раундів для забезпечення стійкості до атак на основі обраного шифрованого тексту для максимально незбалансованої мережі Фейстеля складає до $2^{n(1-\epsilon)}$ кількості запитів атак [31]. Це була гіпотеза, яка на разі доведена Морісом, Рогвеєм та Стейджерсом у роботі про шифрування зі збереженням формату на малих множинах за допомогою перемішування

Торпа [13]. У пропозиції Національного Інституту Інформаційних Стандартів і Технологій США (NIST) Т. Спайс (анг. Spies) [32] описує режим блочного шифрування FFSEM для шифрування на довільній області середнього розміру. Схема містить у собі збалансовану мережу Фейстеля. **Проблематика**. Для будь-якої достатньої кількості раундів r збалансованої мережі Фейстеля з використанням псевдовипадкової раундової функції, що є виходом з PRF — стійкість лежить в межах до $2^n - 2$ запитів атак (невідомо жодної практичної атаки [33]). Доведення стійкості шифрів випливають з псевдовипадкових функцій (PRF), описуючи стійкість, та переходять до теорії складності. Так як збалансована мережа Фейстеля теоретично є за $2^{n/2}$ кількістю запитів атак на основі обраного шифрованого тексту — такий підхід приречений, тому що криптоаналітик може провести $q = 2^{\theta+n/2}$ атак на основі обраного шифрованого тексту для певного $\theta \geq 0$. Тоді для стійкості такої схеми знадобиться чимало раундів, щонайменше $r = 2^{\theta+1}$ [13].

Атака на збалансовану мережу Фейстеля. У роботі Моріса, Рогвея та Стейджерса [13] розглянуто атаку на збалансовану мережу Фейстеля. Це узагальнює відповідь на питання, чому не варто використовувати шифри для забезпечення конфіденційності повідомлень малого розміру, які використовують збалансовану мережу Фейстеля. **Принцип атаки.** Атака з q різних, але відмінних запитів атак на основі обраних відкритих текстів. На вхід зловмисник подає q відмінних відкритих текстів $x_1, ..., x_q, q \in \{0, 1\}^n$ та отримує відповіді: $y_1, ..., y_q \in \{0, 1\}^n$ (відповідні шифровані тексти). Послідовність раундових функцій $f_1, ..., f_r$: $\{0, 1\}^{n/2} \to \{0, 1\}^{n/2}$, які відповідні парам запит/відповідь.

Для того, щоб знизити ефективність зловмисника, потрібно звернути увагу на випадкову перестановку ξ: $\{0, 1\}^n \to \{0, 1\}^n$ — це $N!/(N - q)!$ можливих кортежів $(x_1, y_1), ..., (x_q, y_q)$, які зловмисник може отримати, де $N = 2^n$. n-бітова r-раундова збалансована мережа Фейстеля охоплює

максимум $2^{r\cdot(n/2)\cdot 2^{n/2}}$. Автори [13] отримали оцінку переваги зловмисника:

$$1 - 2^{r\cdot\cdot 2^{n/2}} \cdot N!/(N-q)!$$

Приклад 2.1. Збалансована мережа Фейстеля. Якщо $n = 30$ та $q = 2^{20}$, знадобиться $r = 64$ раунди збалансованої мережі Фейстеля. Тоді перевага криптоаналітика (зловмисника) не буде значимою (великою); по факту при $r = 63$ раунди, перевага криптоаналітика рівна 1 (а саме $1 - 10^{-10000}$).

Щоб переконатися, використаємо отриманий результат авторів [13], які показали що знадобиться $r = 2^{\theta+1}$ раундів для стійкості при $q = 2^{\theta+n/2}$ атак підібраного шифрованого тексту визначивши $r = 2^{\theta}$. Тоді:

$$2^{n/2} \leq 2^n - q = 2^n - 2^{\theta+n/2}, \text{ для } n \geq 2,$$

$$2^{r\cdot(n/2)\cdot 2^{n/2}} = (2^{n/2})^q \leq 2^n \cdot 2^{n-1} \cdot \ldots \cdot (2^n - q + 1) = N!/(N-q)!$$

$$\leq (2^n)^q = 2^{2r\cdot(n/2)\cdot 2^{n/2}}.$$

Можна зробити висновок, що небажано використовувати збалансовану мережу Фейстеля в рамках задачі шифрування зі збереженням формату на невеликій множині.

Загальні оцінки шифру на основі перемішування Торпа

Перемішування карт, яке виявилося важким для аналізу — це перемішування Торпа. Бен Моріс [16] знайшов загальний метод, що зменшує час перемішування.

Нагадаємо, що розмірність повідомлення $|X| = 2^n = N$ є степенем двійки, $r \geq 1$, $E^{Thorp}_{K,R}(X) = Y$ — шифрування на основі перемішування Торпа з $R = 2rn$ кількістю раундів.

У роботі Рогвея, Моріса та Стейджерса [26] показано, що для неадаптивної атаки на основі обраного відкритого тексту при q запитах зловмисника оцінкою стійкості буде величина:

$$\frac{q}{r+1} \cdot \left(\frac{4 \cdot n \cdot q}{N}\right)^r \tag{2.1}$$

Результат було отримано і доведено, виходячи з наступної властивості перемішування Торпа (див. Зауваження 2.2):

Зауваження. Торп перемішує N карт, подібно властивостям ланцюга Маркова.

Означення 2.1. Ланцюг Маркова — послідовність випадкових подій зі скінченним або зліченним числом виходів, які характеризуються властивістю: при фіксованому значенні, наступне отримане значення не залежить від попереднього.

На скільки відомо, це було доведено вперше для симетричних криптографічних систем.

Використовуючи результат Маурера, П'єтзака і Реннера (анг. Maurer, Pietrzak, Renner) [34], Рогвея, Моріса та Стейджерса [26] можемо зробити висновок, що $4r$ кроків достатньо для того, щоб зловмисник виконав q запитів атак на основі обраного шифрованого тексту. Тоді він матиме перевагу при верхній границі (для виділення ширированого тексту, виходу з $E_{k,R}^{Thorp}(X) = Y$ від дійсно випадкової перестановки):

$$\frac{2 \cdot q}{r+1} \cdot \left(\frac{4 \cdot n \cdot q}{N}\right)^r \tag{2.2}$$

Можна побудувати перемішування на n-бітовому вхідному повідомленні, зробивши $2^{n(1-1/r)}$ кроків, з яких 4r подібні (відповідні) максимально незбалансованій мережі Фейстеля (в схемі Фейстеля раундова функція є рівномірною та випадковою від $(n-1)$-го біту до 1-го). Це значно перевищує збалансовану мережу Фейстеля і може забезпечити очевидну перевагу, на відміну від стійкості збалансованої мережі Фейстеля.

При застосуванні наведених вище результатів вирішується проблема шифрування на маленькій довжині вхідного повідомлення:

використовуючи блочний шифр, подібний AES, число раундів R стає числом викликів блочного шифру.

У роботах Моріса доведено, що час перемішування для схеми Торпа рівний приблизно кількості кроків, доки всі q = N біт не будуть впорядковані рівномірно — це $\mathcal{O}(\lg^{44} N)$ [15]. Потім, результат був покращений до $\mathcal{O}(\lg^{19} N)$ [29], пізніше до $\mathcal{O}(\lg^4 N)$ [16].

Наслідком [13] Теореми 2.1 для модифікації п'ятикратного прискорення шифрування перемішуванням Торпа є:

Наслідок 2.1. *Обчислення $Enc_p(x)$ або $Dec_p(x)$ оцінюється не більше ніж $R/5$.*

Зв'язок з мережею Фейстеля.

Твердження 2.2. *Перемішування Торпа еквівалентне максимально незбалансованій мережі Фейстеля.*

Моріс довів [13], що для шифру, на схемі перемішування Торпа $E_k^{Thorp}(X)$, при раунді r переміщеня біту з позиції $x \in \{0, ..., N-1\}$ до позиції: $2x + F_k(r, x)$, якщо $x \leq N/2$, інакше до $2(x - N/2) + (1 - F_k(r, x - N/2))$ — еквівалентно максимально незбалансованій мережі Фейстеля (див. рис. 2.2).

Загальні оцінки шифру на основі перемішування Swap-Or-Not.

Для розв'язку задачі шифрування повідомлень невеликої довжини за допомогою схеми шифрування, що базується на перемішування карт Swap-Or-Not, бралися вчені Гранбулан і Порнін (анг. Granboulan and Pornin) [27], які показали, як реалізувати певне перемішування на N картах з часом перемішування $\mathcal{O}(\lg^3 N)$ в просторі $\mathcal{O}(\lg N)$. Але метод здається нераціональним через застосування розширеної арифметики до вибірки з гіпергеометричного розподілу. Стефанов і Ши (анг. Stefanov, Shi) [35] показали, як використати ідею Гранбулана і Порніна [27] (GP). Їх метод слід застосовувати, якщо можлива генерація ключів у просторі

$\hat{\theta}(N)$, де простір $\hat{\theta}(N^{1/2})$ — необхідний для шифрування повідомлень.

Для довжини повідомлень 2^{30} ці припущення здаються не раціональними. При цьому, запропонований підхід показує стійкість шифру до атак на основі обраного відкритого тексту для усіх N запитів, тоді як гарне шифрування у роботі Гранбулана і Порніна [27] доведено з $(1-\epsilon)N$ кількістю кроків. Автори зробили припущення, що Swap-Or-Not працює добре для N запитів атак та r-раундів і час перемішування Swap-Or-Not є швидким, але такого результату в роботі [35] доведено не було.

Після початкової роботи Моріса, Рогвея та Хоанга [17], Рістенпарт і Єльк [36] використали перемішування Swap-Or-Not, замінивши гіпергеометричні вибірки в побудові Гранбулана і Порніна [27]. Це не тільки полегшує побудову конструкції, але також зменшує кількість раундів від $\theta(\lg^3 N)$ до $\theta(\lg^2 N)$. Моріс і Рогвей [37] покращили цей метод так, що число раундів, у середньому, рівне $\theta(\lg(N))$, але у гіршому випадку залишається, як отримали Рістенпарт і Єльк [36], $\theta(\lg^2 N)$.

У роботі Лубі і Ракофа [9], ми можемо переглянути конструкцію Swap-Or-Not, розглядаючи її складові частини як рівномірно випадкові. Використовуючи узагальнений алгоритм Swap-Or-Not, що продемонстрований у розділі 2.1, значення K_i та F_i обираються рівномірно. Хоанг, Моріс та Рогвей [17] отримали оцінку переваги зловмисника, що проводить атаки обраного шифрованого тексту. Зловмисник отримує оракул $E_{K,F}^{Swap-or-not}(\cdot)$ для випадкового K,F та оракул його інверсії $(E_{K,F}^{Swap-or-not}(\cdot))^{-1}$. Тобто, криплоаналітику надається випадкова рівномірна перестановка $\xi\colon [N] \to [N]$ з оберненою перестановкою $\xi^{-1}(\cdot)$. Максимальний показник над усіма запитами атак q, перевага криптоаналітика складає [17]:

$$\frac{8 \cdot N^{3/2}}{r+4} \cdot \left(\frac{q+N}{2N}\right)^{1+r/4} \tag{2.3}$$

Приклад 2.2. Як простий чисельний приклад, Swap-Or-Not шифрує

64-бітові рядки за 1200 раундів. Випадкова раундова функція надає максимальну перевагу атаками обраного шифрованого тексту менше ніж 10^{-10}, навіть якщо зловмисник може зробити 2^{63} атак.

Хоча кількість раундів, очевидно, велика, жодна інша конструкція не може забезпечити порівнянну гарантію досягнення стійкості навіть тоді, коли q близьке до N.

Блек і Рогвей [7] продовжували роботу над шифруванням зі збереженням формату на невеликій множині. Вони запропонували декілька підходів, найбільш ефективні з яких узагальнюють класичний результат Лебі та Ракофа [9] у застосуванні збалансованої мережі Фейстеля. Це забезпечує стійкість, але лише до $q = 2^{n/4}$ запитів атак обраних шифрованих текстів.

Приклад 2.3. Якщо використовувати збалансовану мережу фейстеля для шифрування повідомлень довжини $n = 30$, існує всього $q \approx 128$ шифротекстів.

Моріс, Рогвей та Стейджерс [13] розкрили зв'язок з незбалансованою мережею Фейстеля і довели, що стійкість Swap-Or-Not складає $q \approx 2^{n(1-\epsilon)}$ запитів атак для перестановки Торпа, обернено пропорційно кількості раундів конструкції.

Хоанг, Моріс та Рогвей [38], запропонувавши Swap-Or-Not, довели, що криптографічна реалізація шифру досягає стійкості $q \approx (1-\epsilon)2^n$.

Mix-and-Cut

Автори Єльк та Рістенпарт [36] пропонують новий шифр, використовуючи роботи [17], [38], [13], що узагальнюють просте перемішування. Нове перемішування включає в себе два підходи. Перший — це процедура рекурсивного перемішування, що лежить в основі роботи Гранбулана і Порніна [27] (далі просто GP), в якій на кожному етапі послідовність ділиться на дві половини, потім кожну половину потрібно поділити на ще дві половини, і так далі, рекурсивно, доки не можна буде

ділити на два. GP використовують при рекурсивній гіпергеометричній вибірці.

Натомість Єльк та Рістенпарт [36] використовують ідею для іншої схеми перемішування, щоб забезпечити криптографічно нерозрізнюване перемішування від PRP. Для цього, необхідно використати певну кількість раундів Swap-or-not, достатніх, щоб забезпечити гарне перемішування.

Вхідні дані: N, X, r
Результат: X
цикл *i := 0 до r-1* **виконати:**
  цикл *j := 0 до N-1* **виконати:**
    отримуємо випадковий біт $K[j]$ з множини $\{0, 1\}$;
    отримуємо випадковий біт $B[j]$ з множини $\{0, 1\}$;
    якщо *B[j] = 1* **тоді:**
      переставити позиції X[i] та X[i] $\oplus K$[i];
    кінець
  кінець
кінець
цикл *i := 0 до (N-1)/2* **виконати:**
  $X[i] \in N_0$;
  $X[i + N/2] \in N_1$;
кінець
Mix-and-Cut(N_0);
Mix-and-Cut(N_1);
Об'єднати N_1 та N_2 операцією конкатенації.

Алгоритм 11: Псевдокод алгоритму Mix-And-Cut.

Отриманий шифр забезпечує повну стійкість, використовуючи лише прості операції: Swap-Or-Not може здійснювати два виклики шифру AES за один раунд, але такий підхід вимагає великої кількості кроків,

наприклад, для повідомлення довжини 2^{30} та переваги зловмисника 10^{-10} потрібно 10,000 раундів. Для порівняння, використання Swap-Or-Not потребує 126×10^9 раундів для повідомдення довжини $N = 2^{30}$. На Intel Core i5 з AES-NI, повне застосування Mix-And-Cut має займати менше мілісекунди. Покращений аналіз для Swap-Or-Not або іншого алгоритму (який можна розглядити як PRP) може бути використаний у схемі Mix-And-Cut для збільшення ефективності.

2.3 Схеми шифрування зі збереженням формату на малій множині

У попередньому розділі (див. Розділ 2.1) ми розглянули відомі та перспективні шифри, що базуються на перемішуванні карт. Твердження Моріса [16] говорить про те, що шифри на основі карточних схем не поступаються в стійкості відомим блочним шифрам:

Твердження 2.3. *Якщо в шифрі, побудованому на схемі перемішування карт, яке є простим (див. твердження 2.1) раундовою функцією використовувати пересворення обраного блочного шифру (див. означення 1.2), то така конструкція шифрує дані не гірше ніж обраний блочний шифр.*

Тобто, якщо в шифрі, побудованому на простому перемішуванні карт, раундовим перетворенням використовувати перетворення шифру AES (запропонований у розділі 1.1), наприклад, то така конструкція буде стійкою з не гіршими оцінками стійкості, що і блочний шифр AES. Цим самим можна стверджувати, що шифри, які використовують карточні перемішування, не поступаються в стійкості відомим блочним шифрам.

Використання складних схем для шифрування малих множин зі збереженням формату, не завжди успішне. Кількість раундів

шифрування має бути як умога меншим, при гарній стійкості.

У цій роботі ми фокусуємося на випадок, коли вхідні дані мають малий розмір та є степенем двійки. Однак, є випадки шифрування зі збереженням формату для невеликих мнижин вхідних даних, що потребують не двійкової системи обчислення (наприклад, десятична система обчислення для кредитних карт). Можна узагальнювати результати робіт з іншими системами обчислення (що стверджують Рістенпарт і Єльк [36] для схеми Mix-And-Cut). Це узагальнення призводить до трохи слабших меж, ніж у бінарному випадку.

При побудові шифрів, що використовують схеми перемішування карт, необхідно застосовувати елемент випадковості: пригадаємо, для розглянених шифрів на карткових перемішуваннях (див. розділ 2.1) запропоновано застосовувати ключ, як вихід функції з PRF (див. означення 1.6).

У пропозиції Національного Інституту Інформаційних Стандартів і Технологій США (NIST), Т. Спайс [32] описує режим блочного шифрування FFSEM, для шифрування на довільній області середнього розміру. Механізм поєднує в собі використання збалансованої мережі Фейстеля. На разі запровадженими стандартами NIST є FFX, та його нащадки (зараз FFX має декілька модифікацій: FF1, FF3, FF3-1 та інші), які пропонують збереження формату: шифрування на основі збалансованої мережі Фейстеля [3].

Криптографічні потреби, на момент створення FFX, вказували на те, що FFX відповідає задачам криптографії, включаючи захист від неадаптивного відновлення повідомлень, стійкість до атак на основі обраного відкритого тексту і навіть стійкість проти адаптивної атаки на основі обраного шифрованого тексту. Передбачається, що основна раундова функція є хорошою псевдовипадковою функцією (PRF). Хоча, FFX може використовуватися для шифрування символьних рядків довільної довжини, механізм призначений для просторів повідомлень, менших ніж у AES. Для шифрування довших рядків, інші методи

здаються кращими.

Наразі відомі атаки на стандарт NIST для FPE. У роботі Дурака та Вауденая (анг. Durak, Vaudenay) [39], показані атаки на FF3, які створені на 8-раундовому застосуванні мережі Фейстеля. Вони показали атаки на основі обраного відкритого тексту та налаштуванням до формату повідомлень невеликої довжини. Це потребує застосування $\mathcal{O}(N^{\frac{7}{4}+\frac{1}{4\cdot L}})$ запитів атак на основі обраного відкритого тексту і двох налаштувань, витрачаючи $\mathcal{O}(N^5)$ часу, де N^2 — це вхід до схеми мережі Фейстеля і L є параметром в запропонованій атаці [39], який зазвичай встановлюється на $\mathrm{L} = 3$.

Автори формують покрокову атаку [39], щоб зламати FF3, також, вони розробили нову загальну атаку підібраного відкритого тексту на 4-х раундовій мережі Фейстеля і додали її в свою атаку на FF3. Методи розробки 4-х раундових атак є новими різновидом атаки на мережу Фейстеля. У їх атаці обчислюється повне відновлення раундової функції з $N^{3/2}(N/2)^{1/2L}$ запитів атак на основі обраного відкритого тексту і складність часу становить $\mathcal{O}(N^{2+3/L})$.

Дурак та Вауденай використовують 4-раундну атаку, щоб розширити функцію відновлення на декілька раундів. Завдяки загальному і відомому відкритому тексту, 4-раундна атака легко пристосовується до атаки на основі обраного відкритого тексту.

Отже, така атака показує, що ні FF1 з $\mathrm{N} = 7$, ні FF3 з $7 \leq N \leq 10$ (навіть з запропонованими в [39] виправленням) — не гарантують стійкість на 128-бітовому просторі вхідних даних.

У відповідь на аналіз FF3 Дурака і Ваунденая [39], NIST оголосив у квітні 2017 року намір переглянути специфікацію FF3, зменшивши розмір його параметрів для вхідних даних від 64 біт до 48 біт, як це було запропоновано дослідниками у роботі [39] або вивести FF3. В SP 800-38G [2], параметр вхідних даних зменшується на 56 бітів. Переглянута FF3 називається FF3-1.

Висновки до розділу 2

На сьогодні вже реалізовано багато алгоритмів, що базуються на схемах перемішування карт та можуть слугувати розв'язком задачі шифрування зі збереженням формату на невеликій множині. Такі підходи до шифрування забезпечують швидкість реалізації, надійність, конфіденційність інформації та стійкість шифру до атак різного виду. Визначено, що перемішування карт може бути гарною основою для перенесення таких схем на блочні шифри. Карточні шифри не поступаються у стійкості до атак (на основі обраного відкритого тексту та обраного шифрованого тексту) відомим блочним шифрам.

Визначено перемішування Торпа, та показано удосконалену реалізацію з п'ятикратним прискоренням.

Розглянуто три алгоритми шифрування Swap-Or-Not, які мають різні підходи до реалізації, але спільну структуру перемішування.

Більшість звичайних перемішувань, таких як Riffle, залежать від попередніх кроків, або від інших карт у колоді. Перестановка Торпа і Swap-Or-Not — не залежать.

Показано повну стійкість шифру Mix-And-Cut, що використовує рекурсивне перемішування Swap-Or-Not. Такий шифр забезпечує гарну стійкість.

Також, згадано стандарт Інституту Інформаційних Стандартів і Технологій США — FFX та його модифікації, що базуються на збалансованій мережі Фейстеля, яка поступається перед картковими шифрами.

3 ОТРИМАННЯ ОЦІНОК ТА ДЕТАЛЬНИЙ АНАЛІЗ ЗАПРОПОНОВАНИХ ШИФРІВ

У даному розділі виконанано огляд та аналіз нового шифру Mix-And-Coin, що поєднує у собі властивості перемішування Swap-Or-Not [17], ідею RS-схеми [19] та базується на конструкції Mix-And-Cut [36]. Доведення оцінок стійкості такого шифру.

3.1 Mix-And-Coin

Розглянемо нову ідею для карточних шифрів. Ми будемо використовувати для раундового перемішування схему Swap-Or-Not, рекурсивну конструкцію викликів Mix-And-Cut та ідею розподілення на групи з RS-схем.

Використання двох підходів (див. Розділ 2.1) процедури рекурсивного перемішування що лежить в основі роботи Гранбулана і Поріна [27], використання Swap-Or-Not [36], як раундової функції перемішування — це основні херектеристики Mix-And-Cut, але ми додаємо ще одне раундове перетворення, яке додає перестановці ще більшої випадковості.

Слід зауважити, що Mix-And-Cut забезпечує повну безпеку, використовуючи прості операції 2.1. Ми намагаємося отримати меншу кількість проходів для отримання того ж результату, збільшуючи довжину випадкових значень раундової функції. Запропонований алгоритм має наступний вигляд.

Вхідні дані: N, X, r

Результат: X

цикл *i := 0 до r-1* **виконати:**

| **цикл** *j := 0 до N-1* **виконати:**
| | $K[i] \overset{\$}{:=} \{0, 1\}$;
| | $B[i] \overset{\$}{:=} \{0, 1\}$;
| | **якщо** *B[i] = 1* **тоді:**
| | | переставити позиції X[i] та X[i] $\oplus K$[i];
| | **кінець**
| **кінець**

кінець

$C \overset{\$}{:=} \{0\}^{n/2}$ або $\{1\}^{n/2}$;

цикл *i := 0 до N-1* **виконати:**

| **якщо** *C[i] = 0* **тоді:**
| | $X[i] \to N_0$;
| **кінець**
| **якщо** *C[i] = 1* **тоді:**
| | $X[i] \to N_1$;
| **кінець**

кінець

Mix-and-Coin (N_0);

Mix-and-Coin (N_1);

Об'єднати N_1 та N_2 операцією конкатенації.

Алгоритм 12: Псевдокод алгоритму Mix-And-Coin.

Алгоритм шифрування Mix-And-Coin дає новий розв'язок проблеми шифрування, що зберігає формат. Узагальнюючи, йдеться мова про побудову шифру з псевдовипадковою перестановкою (PRP) на довжині повідомлення N, використовуючи значення функцій сімейства псевдовипадкових (PRF). (В AES лише один біт буде використано за

кожний 128-бітовий прохід, випадкова перестановка на 128 бітах скорочується до одного віпадкового перемішування біту, надзвичайно близькго до виходу псевдовипадкової функції з PRF). Як у попередніх роботах [6, 3, 23] ідея мотивована перемішуванням карт і його криптографічною інтерпретацією.

Хоанг, Морріс і Рогауей [17] описали просту (див. Твердження 2.1) перемішування Swap-Or-Not, що добре підходить для шифрування на невеликій множині. Після завершення всіх раундів кінцеве значення входу X є результатом перестановки. Автори показали, що часу $\mathcal{O}(\lg N)$ достатньо, щоб отримати шифр, що виглядає однорідним для зловмисника, який робить $q < (1 - \epsilon)N$ запитів атак. Але коли q наближається до N, потрібно більше раундів, і, зрештою, аналітик отримує результат.

Рістернпарт і Єльк в результаті роботи отримали результат, використавши побудоване Хоангом, Моррісом і Рогауєм перемішування Swap-Or-Not[17], що назвали **повною безпекою** для q = N запитів (див. Розділ 2.1). Вони впровадили нову конструкцію (Ristenpart and Yilek's Icicle [36]), що спочатку змішує вхідну множину, використовуючи деяке (назвемо **внутрішнє**) перемішування. Потім множина ділиться на дві частини і рекурсивно перемішується так само. Автори пояснюють: якщо внутрішнє перемішування є гарним псевдовипадковим розділювачем (анг. pseudorandom separator (PRS)), потім побудована перестановка дозволить досягти повної безпеки. Перестановка являється хорошим псевдовипадковим розділювачем (PRS), якщо після перемішування (невпорядкований) набір вхідної множини, не відрізняється від рівномірного розбиття множини на два набори рівних розмірів.

Рістенпарт і Єльк застосовують конструкцію Icicle до Swap-Or-Not і комбінацію, яку вони називають Mix-And-Cut. Поєднання забезпечує повну безпеку для $\theta(\lg^2 N)$ раундів. Коли раундова функція реалізується за допомогою викликів AES, Mix-And-Cut створює шифр на N точок, досягаючи повної безпеки $\theta(\lg^2 N)$. Поки повну безпеку досягають інші

прості (див. Твердження 2.1) перестановки [6, 16, 29], Mix-And-Cut, виконується набагато швидше.

Алгоритм Mix-And-Coin також використовує перемішування Swap-Or-Not і може бути описаний рекурсивно наступним чином. Припустимо, ми хочемо перемішати вхідну послідовність X потужністю $N = 2^n$. Якщо N = 1, одиничну послідовність вже перемішано. В іншому випадку, змішати та розділити $N \geq 2$-послідовність потрібно наступним чином:

1. перемішати N карт, використовуючи внутрішню перестановку;

2. згенерувати випадкову послідовність (виходи функції підкидання чесної монети): $C \overset{\$}{:=} \{0, 1\}^N = \{C[0], C[1], ..., C[N-1]\}$. Розділити послідовність X на дві половини таким чином, що для кожного X[i] на позиції i = { C[0], ..., C[N/2 - 1] }, помістити до множини N_0 та X[i] на позиціях i = { C[N/2], ..., C[N - 1] }, $\forall i \in \{0, ..., N\}$, виконати рекурсивно, перемішуючи кожну половину.

Достатньою умовою для досягнення повної безпеки є не складне перемішування множини. Неформально, щоб злегка перемішати послідовність, це означає, що якщо хтось ідентифікує деякі N/2 позиції X, які потім переставляються до інших позицій — така процедура майже рівномірна. Більш формально, перемішування є розподілом невпорядкованого набору на цих позиціях і знаходиться на відстані рівномірно випадкової підмножини карт розміром N/2 [36].

Розглянемо швидкість перемішування Mix-And-Coin. Для раундової перетасовки, що наближається до рівномірного розподілу, нехай $\tau_q^r(N)$ – визначений розподіл після r раундів на деяких q окремих позиціях $(X[0], ..., X[q]) \in [N]^q$ з N, та нехай $\pi_q(N)$ буде розподілом з q вибірок, без заміни в [N]. Нехай $\Delta_{SN}(N, q, r) = ||\tau_q^r(N) - \pi_q(N)||$ — загальна відстань відхилення між двома розподілами. Хоанг, Моріс і Рогауей показують [17], що для Swap-Or-Not виконується наступна теорема.

Теорема 3.1. *(Хоанг, Моріс, Роугвей [13]). Нехай $N = 2^n$, E_{SN} — ідеальна версія блочного шифру Swap-Or-Not з r раундами. Нехай A — перевага криптоаналітика з атаками неадаптивно вибраного відкритого*

тексту, що робить q запитів. Тоді:

$$Adv^{ncpa}_{E_{SN*}}(A) = \Delta_{SN}(N,\, q,\, r) \leq \frac{2 \cdot N^{3/2}}{r+2} \cdot \left(\frac{q+N}{2N}\right)^{1+r/2}.$$

Наслідок 3.1. *Припускаючи навіть значення N, якщо q=N/2 в даному рівнянні (теорема 3.1) дає:*

$$\Delta_{SN}(N,\, N/2,\, r) \leq N^{3/2} \cdot \left(\frac{3}{4}\right)^{r/2},$$

тоді як $\Delta_{SN}(N,\, N/2,\, r) \leq \epsilon$, *якщо:*

$$\frac{3}{2}\lg(N) + \frac{r}{2}\lg(3/4) \leq \lg(\epsilon),$$

що буде виконуватись, якщо:

$$r \geq \frac{\lg(\epsilon) - \frac{3}{2}\lg(N)}{\frac{1}{2}\lg(3/4)},$$

$$\in \theta(\lg(N) - \lg(\epsilon)).$$

Тепер, можемо оцінити час шифрування для Mix-And-Coin. Для цього позначимо $Time_{SN}(N,\, q,\, \epsilon)$ — мінімальне число r раундів таке, що $\Delta_{SN}(N, q, r) \leq \epsilon$ для Swap-Or-Not. Нехай $Time_{SN}(N,\, \epsilon) = Time_{SN}(N,\, N,\, \epsilon)$ — час, щоб змішати всі N в межах ϵ для Swap-Or-Not, тоді час змішування $Time_{MC}(2^n,\, \epsilon)$ алгоритму Mix-And-Coin в межах ϵ визначений наступним чином:

Твердження 3.1.

$$Time_{MC}(2^n,\, \epsilon) \in \theta(\lg^2(N) - \lg(N) \cdot \lg(\epsilon)).$$

Доведення.

$$Time_{MC}(2^n,\, \epsilon) = \sum_{i=1}^{n} T_{SN}(2^i,\, 2^{i-1},\, \epsilon/n) \leq$$

$$\leq \sum_{i=1}^{n}(7.23 \cdot i - 4.82 \cdot \lg(\epsilon/n)) \text{ (з Наслідку 3.1)}$$

$$\leq 14.46 \cdot n^2 + 4.82 \cdot n \cdot \lg(n) - 4.82 \cdot n \cdot \lg(\epsilon)$$

$$\in \theta(\lg^2(N) - \lg(N) \cdot \lg(\epsilon)).$$

□

Інтерпретація конструкції MC (Mix-And-Coin) може шифрувати n-бітові рядки з будь-якою фіксованою відстанню ϵ, використовуючи $\theta(n)$ кроків з $\theta(n)$ кількістю раундів так, що загальна кількість раундів складатиме $\theta(n^2)$. Раундові функції передбачають однорідність та незалежність. Якщо замінити їх на PRF, ми перетворюємо PRF в PRP на множині $\{0, 1\}^n$ з $\theta(n^2)$ викликами, досягаючи повної безпеки для обмеженої кількості раундів.

Г-псевдовипадковий розділювач (Г-PRS). Для отримання повної безпеки необхідно реалізувати дійсно псевдовипадкову перестановку, що в свою чергу може бути отриманим із застосування псевдовипадкового розділювача та використанням псевдовипадкових функцій. У роботі Рістенпарта і Єлька визначений зв'язок:

$$PRF \xrightarrow{(1)} PRS \xrightarrow{(2)} PRP,$$

(1) — отримуємо за допомогою простої леми 3.1;

(2) — можна одержати за допомогою конструкції бурульки.

Визначимо Г-PRS, запропонований Рістенпартом і Єльком:

Означення 3.1. Стійкий Г-PRS — це така перестановка Z на $\{0, 1\}^n$, що немає обчислювально обмеженого зловмисника, який зміг би виділити перший $\gamma = \log\Gamma$ біт вихідної, випадково-обраної функції $\{0, 1\}^n \to \{0, 1\}^\gamma$.

Стійкість Г-PRS може бути доведена, як стійкість PRF з діапазоном розміру γ біт з використанням методів, які вперше використовувалися

для аналізу стійкості PRF отриманої з PRP (зв'язок Г-PRS та PRP був досліджений та доведений Рістенпартом і Єльком за допомогою леми. У цій роботі ми використовуємо вже запропоновану конструкцію бурульки (анг. Icicle) див. рис. 3.1).

На відміну від PRP, Г-PRS не надає гарантій безпеки щодо решти $n-\gamma$ біт його виходу (при $\gamma < n$). Це означає, що стійкість Г-PRS строго менше, ніж стійкість PRP, але при $\gamma = n$ стійкості рівні.

Ідея використання розділювачів була запозичена у шифрі GP [27], в алгоритмі ідеального розщеплення послідовності на дві половини з використанням рекурсивної вибірки з гіпергеометричного розподілу.

Фіксована величина γ і блок довчини n, $s = n/\gamma$ [22] використовують правила перестановки Z_1, Z_2, ..., Z_s для побудови стійкої PRP на n бітах. На рисунку зображено (див. Рисунок 3.1), як кожне правило Z_i має стійкість 2^γ-PRS для $n - \gamma(i - 1)$ бітів. Використання правил є необхідним для стійкості: вони гарантують, що PRS на пізніших етапах, забезпечить незалежність для різних зашифрованих текстів.

Застосування Г-PRS на $n - \gamma$ біт виконується наступним чином: γ бітів результату «крапають» до наступного кроку, інша частина додається до виходу за правилом Z_i. Приклад для чотирьох біт можна розглянути на рисунку (див. Рисунок 3.1).

Якщо $\Gamma = 2$, рекурсивну процедуру перемішування, що використовується в шифрі GP, тепер можна розглядати як послідовність етапів створення цілком безпечного шифрування, використовуючи гіпергеометричне 2-PRS.

Лема 3.1. *(Рістенпарт і Єльк [36]). Інверсія E_K^{-1} довільного шифру E_K, стійкого PRP для $(\Gamma-1)N/\Gamma$ запитів це хороший $\Gamma - PRS$ для всіх N запитів.*

Лема показує, що можна отримати повну стійкість Г-PRS, використовуючи довільну конструкцію, що досягає $(1 - \epsilon)N$ PRP стійкості. Зокрема, це означає, що Swap-Or-Not — це максимально стійка 2-PRS для певної кількості раундів, щоб зробити його повністю стійкою

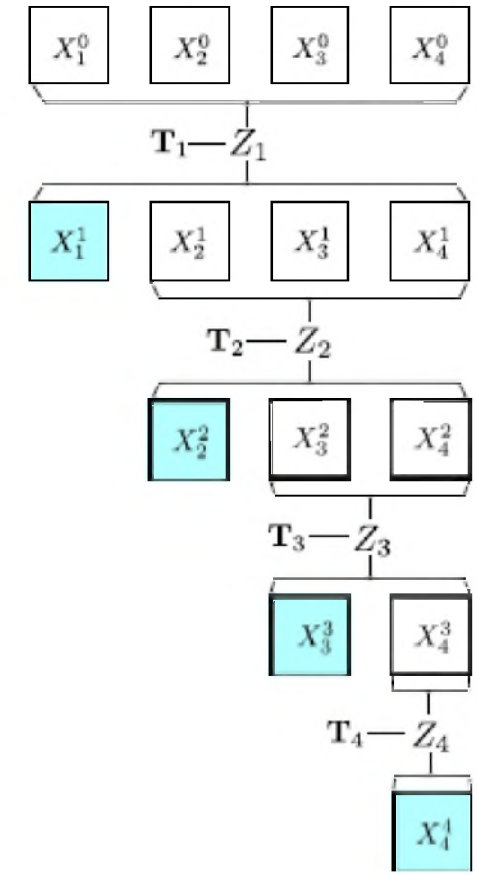

Рисунок 3.1 – Схема шифрування Mix-and-Coin

PRP.

Правила блочного шифрування для конструкції Mix-And-Coin — це сімейство функцій: $\mathcal{E}$: $\mathcal{K} \times \mathcal{T} \times \{0,1\}^n \to \{0,1\}^n$, $\mathcal{K}$ — непорожня множина ключів шифрування, $\mathcal{T}$ — непорожня множина правил шифрування, де для кожного $k \in \mathcal{K}$, $T \in \mathcal{T}$, $E_K(T,\cdot) = E(K,T,\cdot)$ створюють престановку на $\{0,1\}^n$. $E_K^{-1}(T,\cdot)$ — інверсія блочного шифру E. Коли правило $\mathcal{T}$ єдине, ми маємо блочний шифр $E_K(\cdot) = E(K,\cdot)$ та $E_K^{-1}(\cdot) = E^{-1}(K,\cdot)$ (див. Означення 1.2).

Скористаємость визначеннями переваги для атак обраного відкритого тексту та атак підібраного шифрованого тексту.

Твердження 3.2.

$$Adv_E^{CCA}(A) = P\{A^{E_K(\cdot,\,\cdot),\,E_K^{-1}(\cdot,\,\cdot)} \to 1\} - P\{A^{\pi(\cdot,\,\cdot),\,\pi^{-1}(\cdot,\,\cdot)} \to 1\}$$

— перевага зловмисника атаками обраного відкритого тексту.

Для переваги зловмисника атаками підібраного відкритого тексту необхідно визначити доступ до одного з двох різних оракулів, які можуть давати відповідь на запит за одиницю часу для пари векторів $(T_1, ..., T_q)$ та $(M_1, ..., M_q)$. Оракул $E(K, (T_1, ..., T_q), (M_1, ..., M_q))$ обчислює $C_i = E_K^{T_i}(M_i), \forall i$ та повертає шифротекст. Оракул $\pi((T_1, ..., T_q), (M_1, ..., M_q))$ обчислює $C_i = \pi(T_i, M_i)$ для випадкового правила перестановки π та повертає результат. Тепер можна визначити перевагу атак зловмисника неадаптивно обраними відкритими текстами:

Твердження 3.3.

$$Adv_E^{NCPA}(A) = P\{A^{E_K(\cdot,\,\cdot)} \to 1\} - P\{A^{\pi(\cdot,\,\cdot)} \to 1\}$$

— перевага зловмисника атаками підібраного шифрованого тексту.

Перша ймовірність перевищує вибір $K \overset{\$}{:=} \mathcal{K}$ та значення підкинутої чесної монети для A і друга ймовірність $\pi \overset{\$}{:=} Perm(\mathcal{T}, n)$.

Лема 3.2. *[36] (Рістенпарт і Єльк [36]). Нехай $\mathcal{E}: \mathcal{K} \times \mathcal{T} \times \{0,1\}^n \to \{0,1\}^n$ та $N = 2^n$. Нехай A - перевага зловмисника атаками обраного шифрованого тексту, він здійснює атаки для ω різних правил шифрування. Тоді, перевага атаками обраного відкритого тексту B справедлива нервність:*

$$Adv_E^{CCA}(A) \leq Adv_E^{CPA}(B)$$

Тим не менш, B здійснює ωN запитів при цьому, використовує часу як A додати $\mathcal{O}(N\omega \log(\omega N))$.

Псевдовипадковий розділювач. Конструкція Бурульки. Правило перестановки для розділювача виглядає наступним чином:

$Z\colon \mathcal{K} \times \mathcal{T} \times \{0,1\}^n \to \{0,1\}^n, \forall K \in \mathcal{K}, T \in \mathcal{T}, Z(K,T,\cdot)$ — це перестановка з інверсією $Z^{-1}(K,T,\cdot)$. Позначимо $RegFunc(\mathcal{T}, n, \gamma)$ сукупність усіх таких функцій: $f\colon \mathcal{T} \times \{0,1\}^n \to \{0,1\}^\gamma$, де $f(T,\cdot)$ регулярна для довільного $T \in \mathcal{T}$.

Оскільки в Mix-And-Coin ми використовуємо аналогічний Г-PRS, як і Єльк з Рістенпартом для Mix-And-Cut [36], тоді визначимо перевагу для шифру з Г-PRS для запитів атак обраного шифротексту.

Твердження 3.4.

$$Adv_Z^{\Gamma\text{-}PRS}(A) = P\{A^{Z(K,\,\cdot\,,\,\cdot)[\gamma]} \to 1\} - P\{A^{\hat{p}(\cdot,\,\cdot)} \to 1\}$$

Перша ймовірність більша за ймовірність отримання послідовності ключа випардковим чином ($K \overset{\$}{:=} \mathcal{K}$), а друга ймовірність більша за отримання усіх налаштувань T випадковим чином ($\hat{p} \overset{\$}{:=} RegFunc(\mathcal{T}, n, \gamma)$). Оракул $Z(K,\ \cdot\ ,\cdot)\,[\gamma]$ з входом (T, x) повертає перші γ бітів з $Z(K, T, x)$. Мається на увазі, що q = N запитів може бути зроблено для деякого числа правил ω.

Використання Swap-Or-Not в Mix-And-Coin. Нахай F: $\{0,1\}^k \times \{0,1\}^* \to \{0,1\}^n$ — функція з PRF. Використання Swap-Or-Not у блочному шифрі E: $\{0,1\}^k \times \{0,1\}^t \times \{0,1\}^n \to \{0,1\}^n$ побудованим з F та налаштуваннями, кожен раунд передбачає три виклики F: один для генерації раундового ключа, другий для випадкової перестановки позицій і третій для випадкового розділення на дві множини (визначений алгоритм 12). Ідея, як і в перестановцчі Swap-Or-Not, запозичена у Хоанга, Моріса та Рогвея, вони запропонували отримання випадкового біту на кожному проході для позицій X та $X \oplus K$, викликом псевдовипадкової функції.

Побудова PRP з PRS. Формалізуємо криптографічно конструкцію бурульки $I_{c_{\gamma,n}}(Z)$ [36]. Нехай $\gamma, n \in \mathbf{N}$ — числа у проміжку

$1 \leq \gamma \leq n$ і $\gamma | n$. Нехай $\Gamma = 2^\gamma$, Z_i: $\{0,1\}^{k_i} \times \mathcal{T}_i \times \{0,1\}^{n_i} \to \{0,1\}^{n_i}$, $i \in [1 \ldots n/\gamma]$, де $\mathcal{T}_i = \mathcal{T} \times \{0,1\}^{\omega_i}$ для $\omega_i = (i-1) \cdot \gamma$, $n_i = n - (i-1)\gamma$. Зауважимо, коли $i = 1$, $\mathcal{T}_i = \mathcal{T}$, $n_i = n$. Нехай $Z = \{Z_1, Z_2, ..., Z_s\}$. $I_{c_{\gamma,n}}(Z)$ побудована з налаштованого блочного шифру E: $\{0,1\}^{ks} \times \mathcal{T} \times \{0,1\}^n \to \{0,1\}^n$. $I_{c_{\gamma,n}}(Z)$ використовує $s = n/\gamma$ станів. На першому етапі застосовується до повного n-бітового входу Γ-PRS на n-бітах. Приклад вже розглянений на рисунку (див. рис. 3.1).

Теорема 3.2. *(Рістенпарт і Єльк (1) [36]). Для фіксованих $n \geq \gamma \geq 1$, $N = 2^n$, $\Gamma = 2^\gamma$. Нехай наш шифр E: $\{0,1\}^k \times \mathcal{T} \times \{0,1\}^n \to \{0,1\}^n$ це налаштований блочний шифр та E^{-1}: $\{0,1\}^k \times \mathcal{T} \times \{0,1\}^n \to \{0,1\}^n$ — його інверсія. A — Γ-PRS перевага з q запитами для ω різних налаштувань (Визначення 3.4). Тоді для переваги зловмисника B атаками підібраного відкритого тексту буде виконуватись нерівність:*

$$Adv^{\Gamma\text{-}prs}_{E^{-1}}(A) \leq Adv^{cpa}_{E}(B)$$

тобто, B створює $\omega \cdot N \cdot (\Gamma - 1)/\Gamma$ запитів атак підібраного відкритого тексту та витрачає $\mathcal{O}(\omega N(\Gamma-1)/\Gamma) \cdot \log(\omega N(\Gamma-1)/\Gamma)$ часу.

Доведення теореми міститься у роботі Рістенпарта і Єлька [36], присвяченій введенню конструкцій, побудованих на Γ-PRS.

Теорема 3.3. *(Рістенпарт і Єльк (2) [36]). Для фіксованих $n \geq \gamma \geq 1$, $\gamma | n$, $Z = \{Z_i\}_{i=1}^s$ — відповідний набір перестановок для $I_{c_{\gamma,n}}(Z)$. Нехай A — стійкість до атак обраного відкритого тексту з q запитами. Тоді, для Γ-PRS переваги для $B_1 \ldots B_s$, використовуючи результат доведення Теореми 3.2, маємо нерівність:*

$$Adv^{cpa}_{I_{c_{\gamma,n}}(Z)}(A) \leq \sum_{j=0}^{s-1} Adv^{\Gamma\text{-}prs}_{Z_i}(B_s)$$

Тобто, кожна перевага B_j, з q запитами до оракула, обчислюється за час

як A додати $\mathcal{O}(jq \cdot Time(Z) + (s-j)q\log(s-j)q)$.

Доведення теореми міститься у роботі Рістенпарта і Єлька [36], присвяченій введенню конструкцій, побудованих на Γ-PRS.

На жаль, вищезазначена границя не дуже корисна, коли $q \geq N-1$. Замість цього будемо використовувати $q = N/2$, це буде означати, що шифр, побудований на перемішуванні зі стійкістю $(1-\epsilon)\cdot 2^n$ (позначимо як E_{SN}), стійкий 2-PRS для $N\omega$ запитів і ω різних налаштувань (у нашому випадку, це використання Swap-Or-Not, така перестановка має зазначену стійкість).

Визначемо правило перемішування для нашого шифру: $Z_{SN} = \{E_{SN}\}_{i=1}^{n}$ — сімейство блочних шифрів Swap-Or-Not, де E_{SN}^i: $\{0,1\}^k \times \mathcal{T}_i \times \{0,1\}^{n_i} \to \{0,1\}^{n_i}$, $n_i = n-(i-1)$ та $\mathcal{T}_i = \mathcal{T} \times \{0,1\}^n$.

Для простоти, слід використовувати однакову кількість раундів r Swap-Or-Not на кожному етапі.

Тепер можна довести теорему (аналогічно доведенню теореми (4) з [36]):

Теорема 3.4. *Нехай* $N = 2^n$, *фіксовані значення* γ, n: $\gamma | n$, $\Gamma = 2^\gamma$. E_{SN} *— блочний шифр з r раундами, що використовує функції F. A — Γ-PRS перевага для N запитів і* ω *різних налаштувань. Тоді, для переваги B виконується нерівність:*

$$Adv_{E_{SN}}^{2-PRS}(A) \leq \frac{2\cdot\omega\cdot N^{3/2}}{r+2}\cdot\left(\frac{3}{4}\right)^{1+r/2} + Adv_F^{PRF}(B)$$

B виконує $2qr\omega$ *проходів та затрачує A додати* $\mathcal{O}(2r\omega N)$ *часу.*

Доведення. Використовуючи Теорему 3.2, верхньою границею переваги зловмисника B' для E_{SN}, який виконує N/2 фіксованих проходів для кожного (адаптивно обраного) T (зауважимо, тут: $E_{SN} = (E_{SN})^{-1}$).

$$Adv_{E_{SN}}^{2-PRS}(A) \leq Adv_{E_{SN}}^{CPA}(B')$$

Перевагу B' явно вказано в доведенні теореми 3.2 [36].

E_{SN} використовує визначену функцію F (випадкову функцію PRF):

$$Adv^{2-PRS}_{E_{SN}}(A) \leq Adv^{CPA}_{E_{SN}(p)}(B') + Adv^{PRF}_{F}(B)$$

де E_{SN} та $E_{SN}(p)$ з $F(K, \cdot)$ замінюємо випадковою функцією p. Зауважимо, що $E_{SN}(p)$ це не E_{SN*}.

Так як шифр $E_{SN}(p)$ еквівалентний у використанні блочному шифру Swap-Or-Not до E_{SN*}, з новим ключем K, для будь-якого правила T і запитами B', для повідомлення X, до кожного екземпляра з E_{SN*} — кількість запитів фіксовані. Тому, можна використовувати аргумент, в якому неодноразово застосовується перевага атак неадаптивно обраного шифрованого тексту для кожного незалежного стану E_{SN*}, щоб отримати:

$$Adv^{CPA}_{E_{SN}(p)}(B') \leq \sum_{i=1}^{\omega} Adv^{NCPA}_{E_{SN*}}(B_i) \leq \frac{2 \cdot \omega \cdot N^{3/2}}{r+2} \cdot \left(\frac{3}{4}\right)^{1+r/2}$$

(остання нерівність використовує доведення Теореми 3.1 для $q = N/2$ (кількість запитів атак, для кожного B_i)).

Зауважимо, що ω визначає налаштування шифру. На відміну від Mix-And-Cut, потужність множини правил ω в Mix-And-Coin є більшою, через ще один виклик (третій) PRF на кожному раунді. □

Тепер слід дослідити стійкість конструкції, коли E_{SN} використовується як основний 2-PRS. $Z_{SN} = \{E_{SN}\}_{i=1}^{n}$ — сімейство SN блочних шифрів, де

$$E^{i}_{SN}\colon \{0,1\}^{k} \times \mathcal{T}_i \times \{0,1\}^{n_i} \to \{0,1\}^{n_i},$$

$$n_i = n - (i-1),\ \mathcal{T}_i = \mathcal{T} \times \{0,1\}^{i-1}$$

. Позначимо це як

$$I_{c_{1,n}}(Z_{SN})\colon \{0,1\}^{kn} \times \mathcal{T} \times \{0,1\}^{n} \to \{0,1\}^{n}.$$

Перевизначивши блочний шифр на n бітів з простором налаштувань $\mathcal{T}$ побудованих з використанням конструкції Z_{SN}.

Для простоти ми фіксуємо, що це конструкція з однаковим числом раундів r Swap-Or-Not і на кожному етапі (що означає nr раундів Swap-Or-Not для всієї конструкції Z_{SN}). Отримана конструкція називається шифром Mix-And-Coin. Тепер можемо довести наступну теорему (аналогічно доведенню теореми (5) з [36]):

Теорема 3.5. *Нехай $N = 2^n$ та $I_{c_{1,n}}(Z_{SN})$ — це конструкція, описана вище, яка виконує r раундів для кожного виклику i-того виклику (Swap-Or-Not) E^i_{SN}. Кожен виклик E^i_{SN} використовує ту саму визначену функцію F з PRF (зауважимо, з різними ключами). Нехай A — перевага зловмисника для атак обраного відкритого тексту, що робить N запитів для різних ω налаштувань. Тоді, для переваги B, розрізнення виходу з конструкції шифрування Mix-And-Coin від виходу з деякої псевдовипадкової функції (PRF), можна довести наступне:*

$$Adv^{CPA}_{I_{c_{1,n}}(Z_{SN})}(A) < \frac{7 \cdot \omega \cdot N^{3/2}}{r+2} \cdot \left(\frac{3}{4}\right)^{1+r/2} + n \cdot Adv^{PRF}_F(B)$$

B виконує $2rnN\omega$ запитів та виконує обчислення за час $A + \mathcal{O}(2rnN\omega)$.

Доведення. Скористаємось теорамами 3.3 та 3.4, що нам дає:

$$Adv^{CPA}_{I_{c_{1,n}}(Z_{SN})}(A) \leq \sum_{i=1}^{n} Adv^{2-PRS}_{E^i_{SN}}(B_i)$$

$$\leq \sum_{i=1}^{n} 2 \cdot \omega \cdot 2^{i-1} \cdot N_i^{3/2} \cdot (r+2)^{-1} \cdot (3/4)^{1+r/2} + \sum_{i=1}^{n} Adv^{PRF}_F(B_i)$$

де $N_i = 2^{n-(i-1)}$ (рекурсивно ділимо множину N навпіл, за правилом, для кожного раунду), також використали те, що кількість налаштувань для кожного E^i_{SN} — $\omega \cdot 2^{i-1}$ (зауважимо, до множини правил ω відноситься правило розділення множини на половини). По-перше, маємо перевагу B для обраного $j \overset{\$}{:=} [1...n]$ (ймовірність вибрати правильне j — рівномірна і

незалежна), тобто для B_j можна визначити:

$$\sum_{i=1}^{n}(B_i) \leq n \cdot Adv_F^{PRF}(B)$$

Повертаючись до аналізу першої суми:

$$\sum_{i=1}^{n} 2^{i-1} \cdot N_i^{3/2} = \sum_{i=1}^{n} 2^{i-1} \cdot 2^{n-(i-1)} \cdot 2^{(n-(i-1))/2} =$$

$$= 2^n \sum_{i=1}^{n} 2^{i/2} = 2^n \cdot \frac{2^{(n+1)/2} - \sqrt{2}}{\sqrt{2} - 1} < 3.5 \cdot N^{3/2}$$

□

3.2 Атака на шифр Swap-Or-Not (блочне змішування)

Твердження 3.5. *Шифр Swap-Or-Not (визначений у розділі 2.1) з раундовою функцією L:*

$$L_i \bigodot \hat{X} = L_i[0]\,\hat{X}[0] \oplus L_i[1]\,\hat{X}[1] \oplus \ \ldots \ \oplus L_i[n-1]\,\hat{X}[n-1]$$

— не стійкий до атак з обраним відкритим текстом.

Доведення. Розглянемо шифр Swap-Or-Not з кількістю раундів r. Для вхідного повідомлення X, довжина якого N, довжина раундових ключів — фіксована: $K_i = \{0,1\}^N, i \in \{0, ..., r-1\}$, отже ключ $K = \{K_0, K_1, ..., K_{r-1}\}$ фіксований. Довжина додаткових таємних значень, обчислених функцією L_i для кожного раунду i — також фіксована.

Для довільних значень $\{t^0, ..., t^{r-1}\} \in \{0,1\}^{r-1}$ позначимо $\bigoplus_{t_i=1,\ 1\leq i\leq r} k^i$ через $\delta^k_{t^1 \ldots t^r}$.

Через Δ_k позначимо множину $\bigcup\limits_{t^0,\,\ldots,\,t^{r-1}\in\{0,\,1\}^{r-1}} \{\delta^k_{t^1\ldots t^r}\}$.

Оскільки раундова функція не змінює X, або використовує операцію $\oplus$ з раундовим ключем, то результатом раундового перетворення i-го раунду, $0 \leq i \leq r-1$, для значення $X \in \{0,\,1\}^n$ є одне зі значень X або $X \oplus K_i$.

Результатом шифрування Swap-Or-Not відкритого тексту $X \in \{0,\,1\}^n$ при фіксованому значенні ключа K є один з елементів множини $\bigcup\limits_{\delta^k\in\Delta_k} \{\delta^k \oplus x\}$.

Потужність множини Δ_k не перевищує значення 2^{r-1}.

Отже, при фіксованому значенні раундового ключа K_i можна згенерувати певну кількість відкритих текстів випадковим чином та обчислити результат операції $\oplus$ з отриманим шифротекстом, щоб отримати один з елементів множини Δ_k.

Після отримання достаньої кількості елементів множини Δ_k можна розв'язати відповідну систему лінійних рівнянь та відновити невідомі значення раундових ключів $K_0,\ \ldots,\ K_{r-1} \in \{0,\,1\}^n$.

Після цього, відповідним чином, обираючи значення відкритих текстів можна відновити побітово невідомі значення таємних параметрів $L_0,\ \ldots,\ L_{r-1} \in \{0,\,1\}^n$, оскільки операція шифрування при фіксованому значенні ключа K є лінійною відносно кожного з бітів.

□

Висновки до розділу 3

Узагальнено схему шифрування Mix-And-Cut та запропоновано новий шифр на основі Mix-And-Cut. Обчислені оцінки стійкості нового шифру. Показано, як отримати псевдовипадкову перестановку з псевдовипадкової функціїї за допомогою псевдовипадкового розділювача, що і використано для нового шифру.

Також, запропонована нова атака на шифр Swap-Or-Not, що уточнює оцінки стійкості цього шифру.

ВИСНОВКИ

Шифрування зі збереженням формату, являє собою комплекс заходів які забезпечують конфіденційність, цілісність даних, при цьому, таке шифрування може зберегти формат відкритого тексту у зашифрованому. Показано, чому варто використовувати схеми та алгоритми, які зберігають довжину повідомлення або формат.

У роботі визначені відомі карткові перемішування, які можна використовувати для схем шифрування зі збереженням формату, в часності, для шифрування повідомлення невеликої довжини.

На сьогодні вже реалізовано багато алгоритмів, що базуються на схемах перемішування карт та можуть слугувати розв'язком задачі шифрування зі збереженням формату на невеликій множині. Такі підходи до шифрування забезпечують швидкість реалізації, надійність, конфіденційність інформації та стійкість шифру до атак різного виду. Визначено, що перемішування карт може бути гарною основою для перенесення таких схем на блочні шифри. Карточні шифри не поступаються у стійкості до атак (на основі обраного відкритого тексту та обраного шифрованого тексту) відомим блочним шифрам.

Визначено перемішування Торпа, та показано удосконалену реалізацію з п'ятикратним пррискоренням.

Розглянуто три алгоритми шифрування Swap-Or-Not, які мають різні підходи до реалізації, але спільну структуру перемішування.

Більшість звичайних перемішувань, таких як Riffle, залежать від попередніх кроків, або від інших карт у колоді. Перестановка Торпа і Swap-Or-Not — не залежать.

Узагальнено схему шифрування Mix-And-Cut та запропоновано новий шифр на основі Mix-And-Cut. Обчислені оцінки стійкості нового шифру. Показано, як отримати псевдовипадкову перестановку з псевдовипадкової функціїї за допомогою псевдовипадкового розділювача, що і використано

для нового шифру.

Також, запропонована нова атака на шифр Swap-Or-Not, що уточнює оцінки стійкості цього шифру.

БІБЛІОҐРАФІЯ

1. Shannon E., Weaver W. The Mathematical Theory of Comunication. — 1964.

2. Dowkin Morris. Recommendation for Block Cipher Modes of Operation: Methods for Format-Preserving Encryption. — 2016. — Access mode: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-38G.pdf.

3. Mihir Bellare Phillip Rogaway Terence Spies. The FFX Mode of Operation for Format-Preserving Encryption. — 2010. — Access mode: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.304.1736&rep=rep1&type=pdf.

4. Dworkin Morris. Recommendation for Block Cipher Modes of Operation: Methods for Format-Preserving Encryption. — 2016. — Access mode: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-38G.pdf.

5. Eric Brier Thomas Peyrin, Stern Jacques. Stern, BPS: a Format-Preserving Encryption Proposal. — 2010. — Access mode: https://pdfs.semanticscholar.org/0be5/d4c77e333d78ddab5c4bf55d15649a660771.pdf.

6. Baldridge Malcolm Ambler Ernes. Institute for Computer Sciences and Technology National Bureau of Standards: Guidelines for Implementing and Using the NBS Data Encryption Standard. — 1981.

7. Black Jhon, Rogaway Philip. Ciphers with arbitrary finite domains. Topics in Cryptology. — 2002. — P. 114–130. — Access mode: https://eprint.iacr.org/2001/012.pdf.

8. O. Goldreich S. Goldwasser, Micali S. How to construct random functions. — 1986. — P. 792–807.

9. Luby Michael, Rackoff Charles. How to Construct Pseudorandom Permutations from Pseudorandom Functions // SIAM J. Comput. — 1988. — 04. — Vol. 17. — P. 373–386.

10. Aldous D., Diaconis P. Strong uniform times and finite random walks. — 1987. — P. 69–97.

11. Naor Moni, Reingold Omer. On the construction of pseudo-random permutations: Luby-Rackoff revisited. J. of Cryptology. — 1999. — P. 29–66. — Access mode: https://omereingold.files.wordpress.com/2014/10/lr.pdf.

12. Knuth Donald. The Art of Computer Programming. — 1998.

13. Ben Morris Phillip Rogaway Till Stegers. How to Encipher Messages on a Small Domain. — 2009. — Access mode: https://web.cs.ucdavis.edu/~rogaway/papers/thorp.pdf.

14. Morris Ben. The mixing time for simple exclusion. — 2006. — Access mode: https://arxiv.org/pdf/math/0405157.pdf.

15. Morris Ben. The mixing time of the Thorp shuffle. — 2005. — P. 484—-504. — Access mode: https://arxiv.org/pdf/math/0507307v1.pdf.

16. Yuval Yarom Daniel Genkin, Heninge Nadia. Improved mixing time bounds for the Thorp shuffle and L-reversal chain. The Annals of Probability. — 2009.

17. Viet Tung Hoang Ben Morris Phillip Rogaway. An Enciphering Scheme Based on a Card Shuffle. — 2012. — Access mode: https://www.iacr.org/archive/crypto2012/74170001/74170001.pdf.

18. Gilbert E. Theory of shuffling. — 1955.

19. Axel Bacher Oliver Bodini Hsien-Kuei Hwang, Tsai Tsung-Hsi. Generating Random Permutations by Coin-Tossing: Classical Algorithms, New Analysis and Modern Implementation. — 2016.

20. Ressel. Random list permutations in place. — 1992. — P. 271—-275.

21. Brassard G., S. Kannan. The generation of random permutations on the fly. — 1988. — P. 207—-212.

22. Margonda. Perfect shuffle algorithm for cryptografy. — 2014. — Access mode: https://pdfs.semanticscholar.org/3bc9/de8f0d78a6126793045756af4ada2f8b519a.pdf.

23. Ellis John, Fan Hongbing, Shallit Jeffrey. Discrete Mathematics and Theoretical Computer Science. — 2002. — P. 169–180. — Access mode: http://emis.impa.br/EMIS/journals/DMTCS/pdfpapers/dm050111.pdf.

24. Yuval Yarom Daniel Genkin, Heninge Nadia. CacheBleed: A Timing Attack on OpenSSL Constant Time RSA. — 2017. — Access mode: https://eprint.iacr.org/2016/224.pdf.

25. Morris Ben. Rapid mixing of dealer shuffles and clumpy shuffles. — 2014.

26. Mihir Bellare Thomas Ristenpart Phillip Rogaway, Stegers Till. Format-Preserving Encryption. — 2009. — Access mode: https://eprint.iacr.org/2009/251.pdf.

27. Granboulan Louis, Pornin Thomas. Perfect Block Ciphers with Small Blocks. — 2007. — P. 452—-465. — Access mode: https://www.iacr.org/archive/fse2007/45930457/45930457.pdf.

28. Mironov Ilya. (Not So) Random Shuffles of RC4. — 2002. — P. 304—-319. — Access mode: https://eprint.iacr.org/2002/067.pdf.

29. Montenegro Ravi, Tetali Prasad. Mathematical Aspects of Mixing Times in Markov Chains. — 2006. — P. 237—-354.

30. U. Maurer K. Pietrzak. The security of many-round Luby-Rackoff pseudorandom permutations. — 2003. — P. 544—-561.

31. Patarin Jacques. Luby-Rackoff: 7 rounds are enough for $2^{n(1-\epsilon)}$ security. — 2003. — P. 513—-529.

32. Spies Terence. Feistel Finite Set Encryption Mode. — 2008. — Access mode: https://csrc.nist.gov/csrc/media/projects/block-cipher-techniques/documents/bcm/proposed-modes/ffsem/ffsem-spec.pdf.

33. Patarin Jacques. Generic Attacks on Feistel Schemes. — 2008. — Access mode: https://eprint.iacr.org/2008/036.pdf.

34. Ueli Maurer Krzysztof Pietrzak Renato Renner. Indistinguishability amplification. — 2007. — P. 130—-149. — Access mode: https://eprint.iacr.org/2006/456.pdf.

35. Emil Stefanov Elaine Shi. FastPRP: Fast Pseudo-Random Permutations for Small Domains. — 2012. — Access mode: https://pdfs.semanticscholar.org/7639/a9d8bc9cc29d3dcf9e66e582bdb4802e193b.pdf.

36. Thomas Ristenpart Scott Yilek. The Mix-and-Cut shuffle: small-domain encryption secure against N queries. — 2013. — P. 392—-409.

37. Ben Morris Phillip Rogaway. Sometimes-Recurse shuffle: almost-random permutations in logarithmic expected time. — 2014. — P. 311—-326.

38. Hoang Viet Tung, Rogaway Phillip. On Generalized Feistel Networks // Proceedings of the 30th Annual Conference on Advances in Cryptology. — CRYPTO'10. — Berlin, Heidelberg : Springer-Verlag, 2010. — P. 613–630. — Access mode: http://dl.acm.org/citation.cfm?id=1881412.1881455.

39. Betul Durak Serge Vaudenay. Breaking The FF3 Format-Preserving Encryption Standard Over Small Domains. — 2017. — P. 679–707. — Access mode: https://eprint.iacr.org/2017/521.pdf.